COMMUNITY TRACKING IN EVOLVING SOCIAL NETWORKS

SUIVI DES COMMUNAUTÉS DANS LES RÉSEAUX SOCIAUX DYNAMIQUES

par

Ziwei He

Mémoire présenté au Département d'informatique
en vue de l'obtention du grade de maître ès sciences (M.Sc.)

FACULTÉ DES SCIENCES

UNIVERSITÉ DE SHERBROOKE

Sherbrooke, Québec, Canada, July 10, 2018

Le 29 juin 2018

le jury a accepté le mémoire de Ziwei He
dans sa version finale

Membres du jury

Professeur Shengrui Wang
Directeur de recherche
Département d'Informatique

Professeur Belkacem Chikhaoui
Evaluateur interne
Département d'Informatique

Professeur Marie-Flavie Auclair-Fortier
Président-rapporteur
Département d'Informatique

Summary

A social network is a social structure of people related to each other through a common relationship or interest, and the process of investigating it through network and graph theories is called social network analysis. In the last decade, modeling and mining social networks have attracted more and more attention, many researchers are seeking to reveal hidden patterns and their evolutions which can capture interactions between people and groups of people, as well as the associated resources for understanding their behavior.

In our research, we focused on finding and analyzing the evolution of communities in dynamic social networks, which is also known as tracking communities over time. To achieve this, a community-matching strategy is devised, each evolving community will be characterized by a series of significant evolutionary communities. In the social network analysis area, most of the authors just focus on detecting changes (critical events like form, expand, merge, split, etc.) communities may undergo. And they evaluate their algorithms by looking at the number of occurrences of critical events during the whole time period [2], [10], barely focus on tracking community itself. Several methods for tracking communities have been proposed, most of which use however a sequential approach to perform one-to-one community mapping, and the communities are compared in terms of shared-nodes (mostly used Jaccard Coefficient based similarity measure) at only consecutive timestamps. Such one-sided approaches could lead us to a wrong direction of tracking which neglects the social positions of community members and decreases the possibilities of finding the maximum potential evolutions.

To alleviate the limitations mentioned above, we propose a new algorithm for tracking communities. We adopted a two-stage process as follows: first independently detecting communities at each snapshot, then performing many-to-many community-matching on the whole time period with a novel similarity measure to generate a sequence to represent the evolution. The similarity measure we proposed is capable of not only capturing shared-nodes proportion numerically (content similarity), but also the importance of their common members (member quality), and time proximity between communities when we match them. For the tracking strategy, we maximize the pair-wise similarity over all selected matches, which allows for many-to-many mappings between communities across different time steps. The matching is implemented over the entire observation period. It means our method will be able to maximize the potential evolutions we could find. To demonstrate the capacity of the proposed approach to increase the accuracy of tracking, we performed experimental studies.

We carried a comparative study between four existing approaches and our proposed approach for tracking communities to clarify their strength and weakness. In our analysis, we compare the algorithms separately in two main community sets: (1) when groups of users do not overlap and (2) when the groups overlap. After communities are extracted, we implement the five approaches with the same set-up (including same similarity threshold selection method and evaluation criteria) on three different testbeds, extracted from the DBLP, Autonomous System (AS) and Yelp datasets. The communities we can successfully tracked from each approach can lead us to a conclusion: our approach is efficient enough to capture community evolutions over time, and at the same time, has remarkably improved the accuracy of tracking.

Summary

Un réseau social est une structure dans laquelle des personnes sont liées les unes aux autres par le biais d'une relation ou d'un intérêt commun. Le processus d'investigation des réseaux sociaux se basant sur la théorie des graphe et des réseaux est appelée analyse des réseaux sociaux. Au cours de la dernière décennie, la modélisation et le forage des réseaux sociaux ont attiré de plus en plus l'attention. De nombreux chercheurs cherchent à révéler les schémas cachés et les évolutions de ces réseaux, pouvant saisir les interactions entre personnes et groupes de personnes.

Dans notre recherche, nous nous sommes concentrés sur la recherche et l'analyse de l'évolution des communautés dans les réseaux sociaux dynamiques, qui est également connu comme le suivi des communautés au fil du temps. Pour ce faire, une stratégie de correspondance entre les communautés est élaborée, chaque communauté en évolution sera caractérisée par une série de communautés évolutives significatives. Dans le domaine de l'analyse des réseaux sociaux, la plupart des auteurs se concentrent uniquement sur la détection des changements (événements critiques comme la formation, l'expansion, la fusion, la scission, etc.) que les communautés peuvent subir. Et ils évaluent leurs algorithmes en regardant le nombre d'occurrences d'événements critiques pendant toute la période de suivi [2], [10], se concentrent à peine sur le suivi de la communauté en elle-même. Plusieurs méthodes de suivi des communautés ont été proposées, dont la plupart utilisent une approche séquentielle pour effectuer une cartographie des communautés une-à-une, et les communautés sont comparées en termes de noeuds partagés à des instants consécutifs (principalement avec la mesure de similarité basée sur le coefficient Jaccard). De telles approches pourraient nous conduire dans une mauvaise direction pour le suivi des communautés car elles négligent les positions sociales des membres de la communauté et diminue les possibilités de trouver les évolutions potentielles.

Pour pallier les limitations mentionnées ci-dessus, nous proposons un nouvel algorithme de suivi des communautés. Nous avons adopté un processus en deux étapes: premièrement, détecter de manière indépendante les communautés à chaque instant, puis effectuer des correspondances entre plusieurs communautés sur toute la période de suivi avec une nouvelle mesure de similarité pour générer une séquence représentant

l'évolution de celles-ci. La mesure de similarité que nous proposons est capable non seulement de capturer numériquement la proportion des noeuds partagés (similarité du contenu), mais aussi d'évaluer l'importance de leurs noeuds communs (qualité des noeuds) et la proximité temporelle entre les communautés. Pour la stratégie de suivi, nous maximisons la similarité par paire sur toutes les correspondances sélectionnées, ce qui permet des correspondances plusieurs-à-plusieurs entre les communautés au cours des différents instants. La mise en correspondance est miss en oeuvre sur toute la période d'observation. Cela signifie que notre méthode sera capable d'optimiser les évolutions potentielles que nous pourrions trouver. Pour démontrer la capacité de l'approche proposée à augmenter la précision du suivi, nous avons réalisé des études expérimentales.

Nous avons mené une étude comparative entre quatre approches existantes et notre approche proposée pour le suivi des communautés afin de clarifier leurs forces et leurs faiblesses. Dans notre analyse, nous comparons les algorithmes séparément dans deux ensembles de communautés principaux: (1) lorsque les groupes de personnes ne se chevauchent pas et (2) lorsque les groupes se chevauchent. Après extraction des communautés, nous implémentons les cinq approches avec la même configuration (y compris la même méthode de sélection des seuils de similarité et les mêmes critères d'évaluation) sur trois bancs d'essai différents, extraits des bases de données DBLP, Autonomous System (AS) et Yelp. Les communautés que nous pouvons suivre avec succès à partir de chaque approche peut nous mener à une conclusion: notre approche est suffisamment efficace pour capturer les évolutions des communautés au fil du temps et, en même temps, améliore remarquablement la précision du suivi.

Key words: community tracking; similarity measure; evolving networks

Remerciements

I would first like to thank my thesis advisor Professor Shengrui Wang of the Faculty of Science at University of Sherbrooke. The door to Prof. Wang's office was always open whenever I ran into a trouble spot or had a question about my research or writing. He consistently allowed this paper to be my own work, but steered me in the right the direction whenever he thought I needed it.

I would also like to thank my colleagues Etienne Tajeuna who were involved in the validation survey for my research project. Without his passionate participation and input, the validation survey could not have been successfully conducted.

Finally, I must express my very profound gratitude to my parents for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them. Thank you.

Abbreviations

APCC Average Pearson Correlation Coefficient

APNP Average Proportion of Nodes Persisting

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Social network analysis is a process of investigating social structures through the use of networks and graph theory. It entails defining measures in order to capture interactions between people or groups of people, as well as the associated resources for understanding their behavior ([40], [28]). In such studies, social structures are represented as a graph in which individuals are represented by nodes, while the nodes are connected to each other by links that depict the relations among the individuals.

In the domain of social network analysis, tracking the evolution of groups of users within social networks has attracted growing interest from researchers due to the wide variety of application domains, including the mining and analysis of sociological phenomena. For example, in criminology [11], social network methodologies are used to discover and track groups of delinquent individuals over time in order to control them. In the public health field [26], social network strategies can be applied to discover the dynamics of certain subpopulations that are susceptible to a disease, or to predict the early stages of an epidemic. For instance, there has been increased interest in analyzing the formation and evolution of communities of friends in on-line networks, and in understanding individuals (or communities) behavior over time to predict interactions among them [30], [28], [31]. Therefore, tracking the evolution of groups of users is our biggest concern in this thesis.

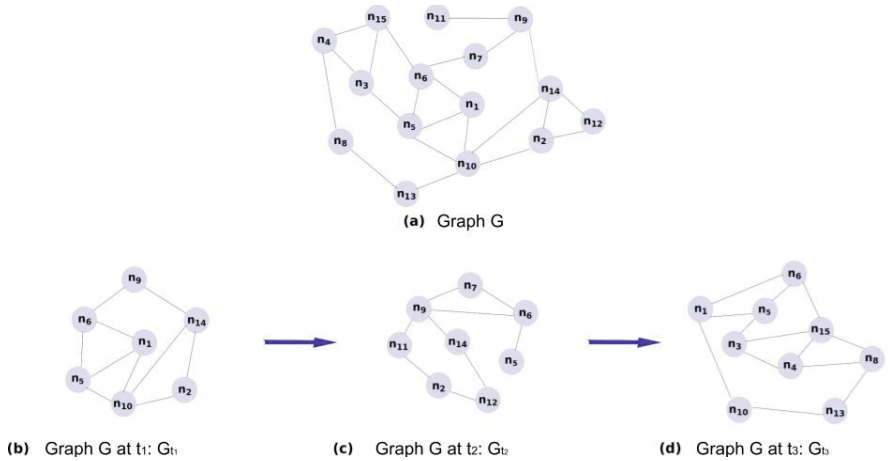


Fig 1.1 – Representation of a static graph (a) as compared to its dynamic representation in (b), (c) and (d). Quoted from [32]

The objective of our study is to track the evolution of communities over time in dynamic social networks. We represent a social network by an undirected and unweighted graph, where the nodes of the graph represent the members of the network, and the edges represent interaction between nodes. We also represent a dynamic social network by a sequence of time snapshots, where each snapshot corresponds to a particular status of the network from a point in time. Groups of users (communities) will be tracked by comparing with other groups at other time steps. We expect to reveal temporal evolution of communities from the tracking results. Below, we propose a novel approach to tracking communities and implementing a comparative study concerning several popular tracking approaches.

1.1 Concept Definitions and Motivation

It is necessary to describe a few general concepts related to social networks and the goal of this thesis.

1. **Dynamic Social Network**:

   Modeling and mining social networks have been a hot topic in the last decade,

with many researchers seeking to reveal hidden patterns and their evolution. To analyze social networks, conventional methods focused on modeling the network as a static graph [27], [23], [25], where the behavior of the individuals is frozen in a snapshot. This type of modeling does not capture the temporal aspect nor the evolution of the network. Recent methods [30], [34], [2], make use of a dynamic graph to model series of networks, where each network corresponds to a particular point in time. Such modeling has proved to be useful in detecting structural changes in the network and in revealing important network information. In Fig 1.1, we give an example which illustrates the difference between static and dynamic social networks.

In this figure, we have two representations of the social network G that has a total number of 15 nodes labeled as n_1, n_2, n_3, ... and n_{15}. In the first case, Fig 1.1(a), we have a global representation of G, whereas in Fig 1.1(b), Fig 1.1(c) and Fig 1.1(d), we have an evolving representation from t_1 to t_3 which relates the dynamic aspect of G. We can note that, in the dynamic case, at each time-stamp, all nodes and links are not always represented. For example, from t_1 to t_2 we can see that nodes n_1 and n_{10} have disappeared, while nodes n_{11} and n_{12} have appeared. In the same way, from t_1 to t_2, we note that links (n_1, n_{10}) and (n_1, n_5) have disappeared from the network, whereas, links (n_2, n_{11}) and (n_2, n_{12}) have appeared in the network.

2. **Evolving Social Network:**

The presence or the absence of nodes from one time to another relates to how likely the graph evolves in time, and such networks are called evolving social networks. We can easily extend this concept to real life, such as adding/removing friends or followers, establishing new collaborations or adding new citations, changing emails/calls graphs over time, etc. Therefore, modeling dynamic social networks as a series of static graphs is useful to detect structural changes in the networks and to reveal important network information. A dynamic social network is modeled as follows. At any time t_i we use the graph structure $g_{t_i} = (V_{t_i}, E_{t_i})$ to represent the snapshot of the social network, where V_{t_i} stands

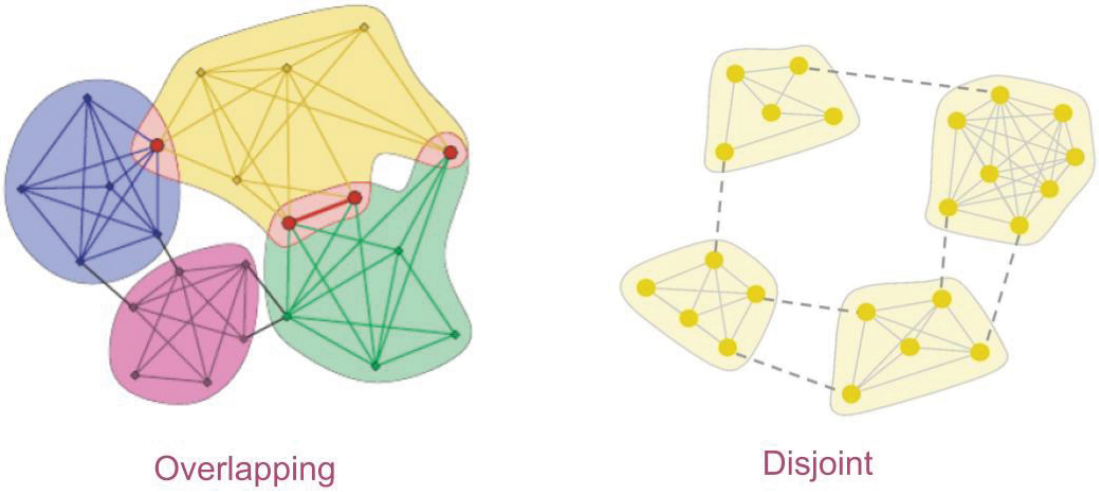


Fig 1.2 – Overlapping Community vs Disjoint Community

for the set of nodes and E_{t_i} the set of edges. Hence, for a duration going from t_1 to t_m, we use the series $G = \{(V_{t_i}, E_{t_i}) \mid 1 \leq i \leq m\} = (g_{t_i})_{1 \leq i \leq m}$ to denote the time evolution of the social network. To detect changes in an evolving social network, two major approaches have been proposed. Some authors use a global approach [30], [28], [31], in which the complete network is tracked over time to observe how nodes and edges behave. Others [34], [2], [33], [5] focus their efforts on tracking communities over time. In this thesis, our focus is on the evolution of communities.

3. **Community:**

   In the social network field, the term "communities" has been defined as groups of users in a way it can be taken as a partition of the social network. Typically, users within a partition are densely connected together, while users in different partitions are not, or only sparsely connected. A simplistic hypothesis assumes disjoint communities where each individual belongs to one and only one group at a time. In the real world, however, an individual may belong to several communities at the same time. In this case, we talk about overlapping communities (see in Fig 1.2).

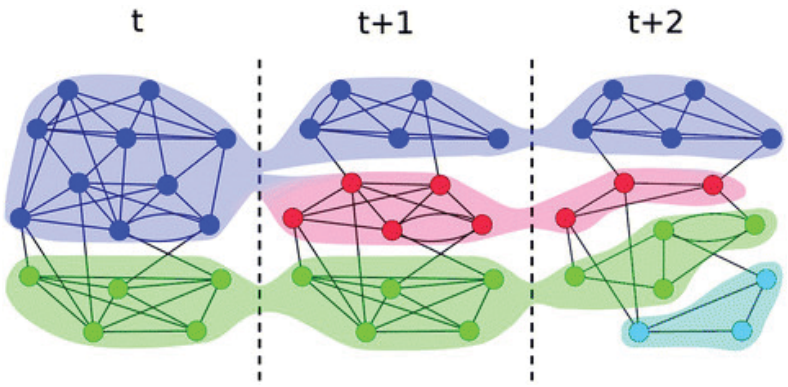


Fig 1.3 – Example of community evolution over three snapshot graphs

A fundamental problem in the analysis of social networks is the detection of communities. There are a lot of ways of doing it, such as minimum-cut method [7], hierarchical clustering [21], modularity maximization [18] and clique-based methods [1]. For each graph g_{t_i}, we detect a set of subgraphs $\left\{C^1_{t_i}, C^2_{t_i}, ..., C^j_{t_i}\right\}$, $j = 1, ..., q_j$, to representing the communities detected at time t_i using a community detection algorithm. Each subgraph $C^j_{t_i}$ is a community of G, with $V^j_{t_i}$ and $E^j_{t_i}$ as its sets of nodes and edges, respectively. With regard to the evaluation of algorithms, to find out which are better at detecting community structure, is still an open question. It must be based on analyses of networks of known structure, and of course, the user's needs. The natural extension of community detection to dynamic networks would be community tracking.

4. **Community Tracking and Evolving Community:**

   In most existing work on analyzing the evolution of community structures in dynamic social networks, the important issues are how to discover transitions or critical events a community may undergo and how to track communities over time. Virtually all of the existing approaches [34], [2], [24], [33], [5], [19], [10] begin by considering the dynamic network under investigation as a series of static snapshot graphs at different time points. Then, using a community detection algorithm to identify community structures at each of these snapshots

independently. The result is a set of communities at each time stamp, which are then matched with communities at other time stamps so that communities can be tracked over time. Finally, they perform community-matching with a similarity measure according to their similarity coefficient. We note that at this point, the authors use different community detection algorithms and different similarity measures to track communities over time. To be more clear, we illustrate the concept of tracking community in Fig 1.3 which shows an example of community evolution over three snapshot graphs.

Communities from different time steps are matched to achieve tracking, the result after matching is a dynamic community sequence. Here we denote by "an evolving community" the sequence of communities tracked, where each community within this sequence indicates the status of the evolving community at a specific time point. For example, in the Fig 1.3 each color stands for an evolving community. It is believed that the sequence can show the completed process of the evolution of a community. For instance, $S_{C^a} = \{C^a_{t_1}, C^a_{t_2}, C^a_{t_3}, ..., C^a_{t_9}\}$ will be the evolution of community C^a from t_1 till t_9. It is worth noting that the communities from a sequence could be identified at two consecutive or non-consecutive timestamps.

5. **Critical Events:**

   From one time to another, an evolving community may change its structure due to the arrival and/or departure of nodes and edges. Hence, given a community evolving in time, it may *expand* because several nodes have joined the community, or *shrink* because several nodes have left. In the same way, the community may *split* into different communities, or several communities may *merge* into one community. We may also observe a community that completely *dissolves* over time. Fig. 1.4 illustrates the different types of events a given community may undergo between two different times t_i and t_j $(t_i < t_j)$. All of these changes are also known as critical events. But our goal of this thesis is tracking communities over time, does not cover the research of critical events that can happen to a community.

1.2 Goal and Organization of Thesis

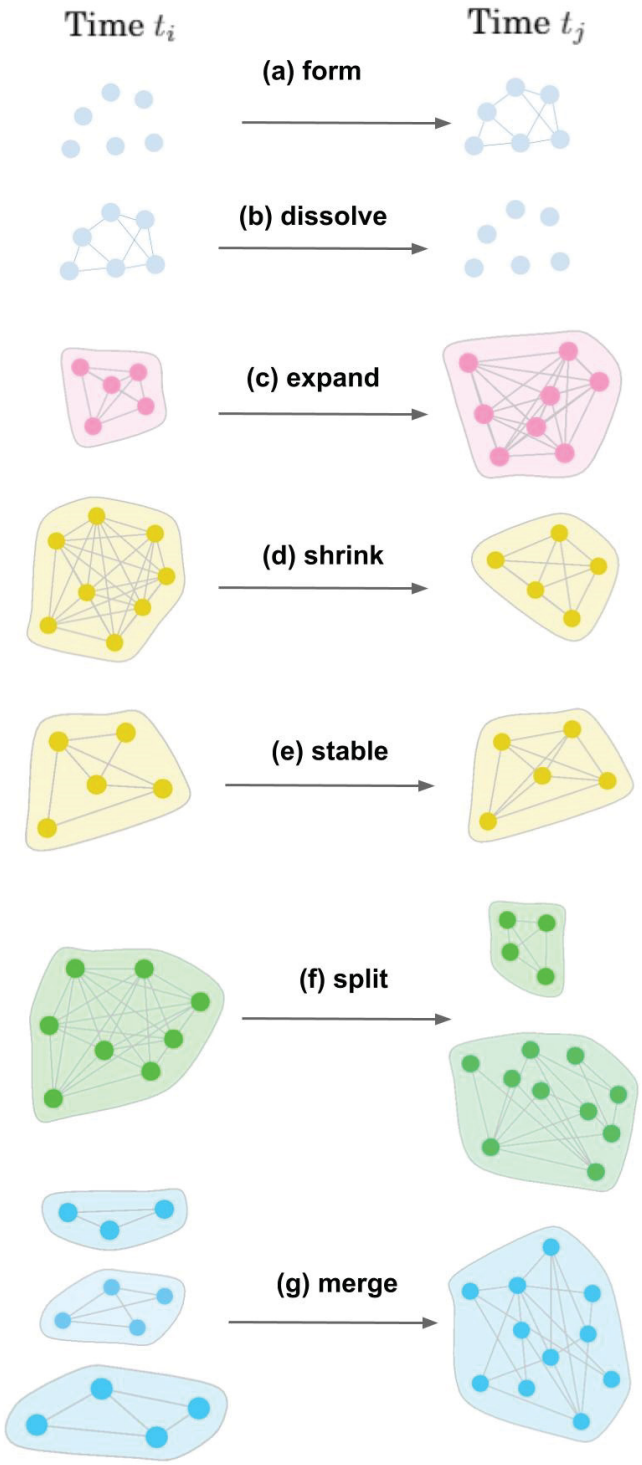
The rest of this thesis is divided into three main chapters, each chapter targeting a precise goal. The chapter 2 investigates the existing approaches for tracking communities over time and detecting the critical events they are susceptible to undergo. This investigation has helped us to better orientate our studies and better target the main existing problems.

In chapter 3, the goal is to establish a methodology to deal with the problems the existing methods have when tracking communities over time. For this purpose, we adopted a new approach with a novel similarity measure to keep the "Nodes Quality", "Time Proximity" and "Content Similarity" concurrently. We divide this chapter into two sections. In the first section, we describe in details about the problems we encountered during our research. In the second section, we introduce in details the steps we used to track communities over time. Firstly, we introduce a new similarity measure that helps in comparing the communities taking consideration their attributes. Then, we introduce a principled approach to automatically identifying the similarity threshold when comparing communities. At the end of the chapter, we

give the algorithm we use to track communities.

In chapter 4, we validate the proposed approach with four other state-of-the-art approaches on three different datasets. The goal is to discover the best algorithm against different type of dataset. It is divided into three main sections. Firstly, we briefly introduce those approaches we are going to demonstrate. Then we describe step by step how the experiments have been done, and give two criteria we adopted to evaluate tracking results. Finally, we present the experimental results with respect to the criteria we set.

Fig 1.4 – Possible events a community may undergo as time evolves from time t_i to t_j



Chapter 2

Related work

In dynamic social networks studies such as analyzing the evolution of community structures, some of the main issues are how to track communities over time and how to discover transitions or critical events a community may undergo. Some authors use a global approach [31, 30, 28, 3] in which the complete network is tracked over time with the aim of predicting the arrival or departure of nodes. This approach permits to reveal the social network properties via the nodes or links activities over time as the appearance/disappearance. Other authors [35, 14, 8, 16] use a local approach in which communities are discovered first and then tracked over time with the aim of tracking them successfully and predicting their next structure. Since we focus our research on a local approach, and tracking communities is our biggest concern, so in what follows, we discuss about local approaches used to track communities over time and also detect critical events they may undergo. Please note that detecting and analyzing critical events is not implemented in this thesis, because exploring critical events is a natural extension of community tracking, and also a future goal of our study, it will still be presented in this chapter.

2.1 Asur et al. [2]

Asur et al. [2] defined an event-based framework to characterize complex behavioral patterns of individuals and communities over time. In their paper, they first detected disjoint community structures for each snapshot of the network at a specific time using the Markov Clustering Algorithm [37]. Then, they defined five events that communities can undergo between any two consecutive snapshots. The events are *stable*, *merge*, *split*, *form*, and *dissolve*. To find these events, they compared two communities at two successive timestamps by investigating the number of nodes shared.

The critical events are defined as follows,

1. *stable*: A community remains stable if, at two consecutive time-stamps, we still have the same nodes. Formally, a community $C_{t_i}^j$ remains stable if there exists at time t_{i+1} another community having exactly the same nodes with $C_{t_i}^j$:

$$V_{t_i}^j \;=\; V_{t_{i+1}}^j$$

.

2. *merge*: Nodes of two communities l and q discovered at a given time t_i, are joined into one community j at time t_{i+1},

$$\frac{|\left(V_{t_i}^l \cup V_{t_i}^q\right) \cap V_{t_{i+1}}^j|}{max\left(|V_{t_i}^l \cup V_{t_i}^q|,\, |V_{t_{i+1}}^j|\right)} \;>\; k\%$$

under the given constraints: $|V_{t_i}^q \cap V_{t_{i+1}}^j| \;>\; \frac{|V_{t_i}^q|}{2}$ and $|V_{t_i}^l \cap V_{t_{i+1}}^j| \;>\; \frac{|V_{t_i}^l|}{2}$

3. *split*: Nodes of a community j at time t_i are split into two communities l and q at the next time point t_{i+1},

$$\frac{|\left(V_{t_{i+1}}^l \cup V_{t_{i+1}}^q\right) \cap V_{t_i}^j|}{max\left(|V_{t_{i+1}}^l \cup V_{t_{i+1}}^q|,\, |V_{t_i}^j|\right)} \;>\; k\%$$

under the constraints: $|V_{t_{i+1}}^q \cap V_{t_i}^j| \;>\; \frac{|V_{t_{i+1}}^q|}{2}$ et $|V_{t_{i+1}}^l \cap V_{t_i}^j| \;>\; \frac{|V_{t_{i+1}}^l|}{2}$

4. *form*: From one time t_i to time t_{i+1}, no more than one nodes in community j is found at time period t_i.

$$\forall C_{t_i}^j \in g_{t_i} \ |V_{t_i}^j \bigcap V_{t_{i+1}}^j| \leq 1$$

5. *dissolve*: Nodes found in a community at time t_i are completely dissolved at time t_{i+1},

$$\forall C_{t_{i+1}}^j \in g_{t_{i+1}} \ |V_{t_i}^j \bigcap V_{t_{i+1}}^j| < 1$$

In their paper, they did not clearly specify an algorithm for tracking communities over time. So the hypotheses for finding critical events were used to track the evolution of communities over time

2.2 Greene et al. [19]

Like Asur et al. in [2], Greene et al. adopted the same strategy to detect critical events the communities may undergo. Note that, in their experiments, rather than using the Markov clustering algorithm to identify the communities, Greene et al. used the Blondel modularity optimization algorithm [6] to identify disjoint communities at different timestamps.

Though the strategy used to identify critical events is similar to that in [2], there are differences in the definition of critical events. One such difference concerns the *dissolve* event. Greene et al. assumed a community observed at a given time t_i to be dissolved if, after $d > 2$ consecutive times, none of its nodes is present in the graph. This rule is used to define if a community is still "alive", in other words, a community will be considered as "dead" if it disappears for more than 2 time stamps speaking of tracking communities.

According to Greene et al., the critical events are defined as below,

1. *shrink*: A community observed at time t_i looses most of its nodes at time t_{i+1};

2. *expand*: A community observed at time t_i gains nodes at time t_{i+1};
3. *split*: A community observed at time t_i is divided into two communities at time t_{i+1};
4. *merge*: Two communities observed at time t_i are joined into one community at time t_{i+1};
5. *form*: A community is formed at time t_i and there is no community similar to this one at all previous time;
6. *dissolve*: A community observed at time t_i is dissolved if during at least the three consecutive time-stamps there is no community similar to this one.

In their approach, Greene et al., identified sequences of communities that represent the evolution of community structures. To this end, at each time they compared the ratio of nodes shared among the communities at consecutive timestamps. Hence, two communities (identified at t_i and t_{i+1}) are aligned in the same sequence if they share at least $k\%$ of nodes according to the Jaccard coefficient as follows,

$$sim(C^k_{t_i},\ C^j_{t_{i+1}}) \ = \ \frac{|V^k_{t_i} \ \bigcap \ V^j_{t_{i+1}}|}{|V^k_{t_i} \ \bigcup \ V^j_{t_{i+1}}|} \ \geq \ k\% \tag{2.1}$$

Note that the condition they imposed for declaring a community dissolved allowed Greene et al. to discover evolving communities at non-consecutive timestamps. However, this evolving discontinuity is influenced by the user-determined parameter d.

It is worth noting that [2] and [19] assume that a community can only be split into two communities, and only two communities can merge into one community in the interval between consecutive times.

2.3 Takaffoli et al. [34]

In their method, Takaffoli et al. used the local community mining algorithm [12] to produce sets of disjoint communities for each snapshot. In contrast to the previous approaches, in which most of the critical events took place only in consecutive

timestamps, Takaffoli et al.'s method is capable of identifying critical events that occur at consecutive and non-consecutive timestamps. The authors can thus track communities evolving in a non-consecutive fashion. The tracking process operates by comparing the communities at different timestamps, using the following similarity measure:

$$Sim(C_{t_i},\ C_{t_j}) = \begin{cases} \frac{|V_{t_i} \bigcap V_{t_j}|}{max(|V_{t_i}|, |V_{t_j}|)} & if \quad \frac{|V_{t_i} \bigcap V_{t_j}|}{max(|V_{t_i}|, |V_{t_j}|)} \ \geq \ k \\ 0 & otherwise \end{cases} \tag{2.2}$$

It is important to note that the *threshold* similarity k is automatically determined. The *threshold* is evaluated using a text-mining approach. Therefore, the authors evaluate their approach on social networks that incorporate content information, such as DBLP and ENRON email dataset [22], where they can exploit the information shared between nodes.

In contrast to the previous approaches, where a community can only split into two smaller communities, Takafolli et al.'s method is able to detect communities splitting into more than two communities. In the same way, it is capable of detecting more than two communities merging into one community.

In their approach they define and formalize critical events as follows,

1. *form*: A community C_{t_i} is formed at time t_i, if there is no community similar to this one at all previous time,

$$\forall C_{t_j},\ j \ < \ i,\ Sim(C_{t_i},\ C_{t_j}) \ = \ 0$$

2. *dissolve*: A community C_{t_i} is dissolved at time t_i if there exist no community similar to one at all further time,

$$\forall C_{t_j},\ j \ > \ i,\ Sim(C_{t_i},\ C_{t_j}) \ = \ 0$$

3. *merge*: A set of communities $S \ = \ \{C^1_{t_i},\ ...,\ C^m_{t_i}\}$ merge at time t_j, $j \ > \ i$ if there exists one community C_{t_j} similar to all of them,

$$\begin{cases} \forall C_{t_i}^r \in S, : \frac{|V_{t_i}^r \bigcap V_{t_j}|}{|V_{t_i}^r|} \geq k \\ \frac{|\left(V_{t_i}^1 \bigcup V_{t_i}^2 ... \bigcup V_{t_i}^m\right) \bigcap V_{t_j}|}{|V_{t_j}|} \geq k \end{cases}$$

4. *split*: A community C_{t_i} is divided at time t_j, $j > i$ if there exist a set of communities $S = \{C_{t_j}^1, ..., C_{t_j}^m\}$ similar to it,

$$\begin{cases} \forall C_{t_j}^r \in S : \frac{|V_{t_j}^r \bigcap V_{t_i}|}{|V_{t_j}^r|} \geq k \\ \frac{|\left(V_{t_j}^1 \bigcup V_{t_j}^2 ... \bigcup V_{t_j}^m\right) \bigcap V_{t_i}|}{|V_{t_i}|} \geq k \end{cases}$$

5. *survive*: A community C_{t_i} survive at time t_j, $j > i$ if there exists a community C_{t_j} similar to it,

$$Sim(C_{t_i}, C_{t_j}) = \frac{|V_{t_i} \bigcap V_{t_j}|}{max(|V_{t_i}|, |V_{t_j}|)}$$

Note that, the event *survive* includes other transition events which are,

(a) *expand*: A community C_{t_i} expands at a further time t_j if this community first of all survives then has more nodes at this last time,

$$Sim(C_{t_i}, C_{t_j}) = \frac{|V_{t_i} \bigcap V_{t_j}|}{max(|V_{t_i}|, |V_{t_j}|)} \; and \; |V_{t_i}| < |V_{t_j}|$$

(b) *shrink*: A community C_{t_i} shrinks at a further time t_j if this community first of all survives then has less nodes at this last time,

$$Sim(C_{t_i}, C_{t_j}) = \frac{|V_{t_i} \bigcap V_{t_j}|}{max(|V_{t_i}|, |V_{t_j}|)} \; and \; |V_{t_i}| > |V_{t_j}|$$

(c) *compact*: A community C_{t_i} becomes compact at time t_j if it survives at this time and gains more edges as follows,

$$Sim(C_{t_i}, C_{t_j}) = \frac{|V_{t_i} \bigcap V_{t_j}|}{max(|V_{t_i}|, |V_{t_j}|)} \; and \; \frac{|E_{t_i}|}{|V_{t_i}|(|V_{t_i}| - 1)} < \frac{|E_{t_j}|}{|V_{t_j}|(|V_{t_j}| - 1)}$$

(d) *diffuse*: A community C_{t_i} diffuses at time t_j if it survives at this time and looses edges as follows,

$$Sim(C_{t_i},\ C_{t_j}) \ =\ \frac{|V_{t_i} \bigcap V_{t_j}|}{max(|V_{t_i}|, |V_{t_j}|)}\ And\ \frac{|E_{t_i}|}{|V_{t_i}|(|V_{t_i}|-1)} > \frac{|E_{t_j}|}{|V_{t_j}|(|V_{t_j}|-1)}$$

2.4 Brodka et al. [10]

In all of the above methods, the authors did not test their approaches on overlapping communities, which might reveal different information. To address this shortcoming, Brodka et al. developed a flexible approach called group evolution discovery (*GED*) which is able to track disjoint and overlapping communities.

Note that in the previous approaches, to detect changes and track the evolution communities may undergo, authors relied on a measure based only on the proportion of nodes shared at two consecutive times. Brodka et al. extended this measure by including a topological metric in their comparison. This similarity measure for locating critical events is used for matching communities speaking of tracking.

$$I(C_{t_i},\ C_{t_{i+1}}) \ =\ \frac{V_{t_i} \ \bigcap\ V_{t_{i+1}}}{V_{t_i}} \ \times\ \frac{\sum_{n\ \in\ V_{t_i}\ \bigcap\ V_{t_{i+1}}} NI_{C_{t_i}}(n)}{\sum_{n\ \in\ V_{t_i}} NI_{C_{t_i}}(n)} \tag{2.3}$$

where $NI_{C_{t_i}}(n)$ reflects the Importance of Node (NI) n within the community C_{t_i}. This measure can be any centrality metric (centrality, social position, degree, etc.). With the added topological metric, their comparison is also able to consider the interrelation among the nodes in a community.

Though the comparison is done at consecutive timestamps, the inclusion effect of their comparison metric helps Brodka et al's method track overlapping and non-overlapping communities. Moreover, within two consecutive time-stamps, they are

able to detect the *form*, *dissolve*, *stable*, *expand*, *shrink*, *merge* and *split* events which are defined as,

1. *stable*: A community C_{t_i} remains stable at time t_{i+1} if there exists a community $C_{t_{i+1}}$ such that,

$$NI(C_{t_i},\ C_{t_{i+1}})\ \geq\ \alpha\ and\ NI(C_{t_{i+1}},\ C_{t_i})\ \geq\ \beta\ and\ |V_{t_i}|\ =\ |V_{t_{i+1}}|$$

2. *shrink*: A community C_{t_i} shrinks at time t_{i+1} if there exists a community $C_{t_{i+1}}$ such that,

$$NI(C_{t_i},\ C_{t_{i+1}})\ \geq\ \alpha\ and\ NI(C_{t_{i+1}},\ C_{t_i})\ \geq\ \beta\ and\ |V_{t_i}|\ >\ |V_{t_{i+1}}|$$

   or

$$NI(C_{t_i}, C_{t_{i+1}})\ <\ \alpha\ and\ NI(C_{t_{i+1}},\ C_{t_i})\ \geq\ \beta\ and\ |V_{t_i}|\ \geq\ |V_{t_{i+1}}|$$

3. *expand*: A community C_{t_i} expands at time t_{i+1} if there exists a community $C_{t_{i+1}}$ such that,

$$NI(C_{t_i},\ C_{t_{i+1}})\ \geq\ \alpha\ and\ NI(C_{t_{i+1}},\ C_{t_i})\ \geq\ \beta\ and\ |V_{t_i}|\ <\ |V_{t_{i+1}}|$$

   or

$$NI(C_{t_i},\ C_{t_{i+1}})\ \geq\ \alpha\ and\ NI(C_{t_{i+1}},\ C_{t_i})\ <\ \beta\ and\ |V_{t_i}|\ \leq\ |V_{t_{i+1}}|$$

4. *split*: A community C_{t_i} splits at time t_{i+1} if there exists a set of communities $S = \{C^1_{t_{i+1}},\ ..., C^m_{t_{i+1}}\}$ such that, $\forall C^j_{t_{i+1}}\ \in S$, $C^j_{t_{i+1}}$ is a shrinkage of C_{t_i}

5. *merge*: A set of communities $S = \{C^1_{t_i},\ ..., C^m_{t_i}\}$ observed at time t_i merge at time t_{i+1} if there exists a community $C_{t_{i+1}}$ such that, $C_{t_{i+1}}$ is an enlargement of each community in S

6. *dissolve*: A community C_{t_i} dissolves at time t_i if there is no community similar to C_{t_i} at time t_{i+1}

7. *form*: A community C_{t_i} is formed at time t_i if there is no similar community at the previous time-stamp.

2.5 Gliwa et al. [16]

Like the *GED* framework used in [10], Gliwa et al. proposed the Stable Group Changes Identification (SGCI) method to track and predict community changes in dynamic social networks. Note that in their approach, they used the *CPM* algorithm [1] to extract overlapping communities at each timestamp of the dynamic social network. However, rather than using the Inclusion metric (Eq 2.3) as in [10], they proposed a modified Jaccard (MJ), given as follows:

$$MJ(C_{t_i}, C_{t_j}) = max\left(\frac{C_{t_i} \bigcap C_{t_j}}{C_{t_i}}, \frac{C_{t_i} \bigcap C_{t_j}}{C_{t_j}}\right) \tag{2.4}$$

where two communities (C_{t_i}, C_{t_j}) are assumed to be similar if $MJ(C_{t_i}, C_{t_j}) \geq 0.5$.

Moreover, they assume that the more there are features the more there can have good results when predicting if a community will undergo a critical event. This is why, they extended the work of [8] by using more topological features such as,

1. *leadership*,
$$L = \sum_{v \in C_{t_i}} \frac{d_{max} - d(v)}{(n-2)(n-1)}$$
with d_{max} the highest degree of nodes in C_{t_i}, $d(v)$ is the degree of the node v in community C_{t_i}, n the number of nodes in C_{t_i}.

2. *density*,
$$D = \frac{\sum_u \sum_v a(u, v)}{n(n-1)}$$
where $a(u, v) = 1$ if there exists a link between the nodes u and v and 0 otherwise.

3. *cohesion*,
$$C = \frac{\frac{\sum_{u \in C_{t_i}} \sum_{v \in C_{t_i}} a(u,v)}{n(n-1)}}{\frac{\sum_{u \in C_{t_i}} \sum_{v \notin C_{t_i}} a(u,v)}{N(N-n)}}$$
with N the number of nodes in the network.

4. *size*,

$$T = |V_{t_i}|$$

2.6 Tajeuna et al. [33]

In order to track communities and extract critical events, Tajeuna et al. first represented each community as a vector (called the *transition probability vector*) indicating the number of nodes shared by the communities over time. Then, they compared the corresponding vectors of different communities. Specifically, given two communities C_{t_i} and C_{t_j} with *transition probability vectors* v_i and v_j, respectively, they calculated the similarity between the two communities as

$$sim(C_{t_i},\ C_{t_j}) = \begin{cases} \sum_{\alpha=1}^{C}\ 2\ \frac{p_{i,\,\alpha} \times p_{j,\,\alpha}}{p_{i,\,\alpha} + p_{j,\,\alpha}} \\ \\ \quad if\ \ \sum_{\alpha=1}^{C}\ 2\ \frac{p_{i,\,\alpha} \times p_{j,\,\alpha}}{p_{i,\,\alpha} + p_{j,\,\alpha}}\ >\ \lambda \\ \\ 0 \quad otherwise \end{cases} \tag{2.5}$$

where λ is the junction point between the two Gamma curves estimated from the non-zero values obtained when scoring the similarity between two transition probability vectors; $p_{i,\,\alpha}$ and $p_{j,\,\alpha}$ are the respective components of vectors v_i and v_j. N_c contains all the communities.

Tajeuna et al. supposed all communities C_{t_j} in $S_{C_{t_i}}$ (the evolution of C_{t_i}) should always share nodes with C_{t_i} such that the Jaccard coefficient exceeds the threshold λ (defined in the previous paragraph), as follows:

$$J(C_{t_i},\ C_{t_j}) = \frac{|V_{t_i}\ \bigcap\ V_{t_j}|}{|V_{t_i}\ \bigcup\ V_{t_j}|}\ >\ \lambda \tag{2.6}$$

Using both of the similarity measures described in Eq 2.5 and Eq 2.6, they defined the evolution of community C_{t_i} as the sequence of sorted communities $S_{C_{t_i}}$ =

2.6. Tajeuna et al. [33]

$\{C_{t_i}, C_{t_{i+\eta}}, ..., C_{t_k}\}$, $t_i < t_k \leq t_m$ such that all communities in $S_{C_{t_i}}$ are similar.

The critical events are defined as follow [33],

— *form*: A community C_{t_i} forms at time t_i if, the proportion of nodes shared by any community $C_{t_\gamma} \in G$, $t_\gamma < t_i$ and community C_{t_i} do not exceed the threshold similarity λ:

$$form(C_{t_i}) = 1 \ \ if \ \ \forall\, C_{t_\gamma} \in G,\ J(C_{t_i},\ C_{t_\gamma}) \leq \lambda$$

— *dissolve*: A community C_{t_i} dissolves at time t_i if, the proportion of nodes shared by any community $C_{t_\theta} \in G$, $t_\theta > t_i$ and community C_{t_i} do not exceed the threshold similarity λ:

$$dissolve(C_{t_i}) = 1 \ \ if \ \ \forall\, C_{t_\theta} \in G,\ J(C_{t_i},\ C_{t_\theta}) \leq \lambda$$

— *shrink*: A community C_{t_i} shrinks, if there exists a community $C_{t_\theta} \in G$, $t_\theta > t_i$, that is smaller than, similar to and shares nodes with C_i:

$$shrink(C_{t_i}) = 1 \ \ if \ \ \exists\, C_{t_\theta} \in G/$$
$$sim(C_{t_i},\ C_{t_\theta}) = 1, \frac{|V_{t_i} \bigcap V_{t_\theta}|}{|V_{t_i} \bigcup V_{t_\theta}|} > \lambda,\ |V_{t_i}| > |V_{t_\theta}|$$

— *expand*: A community C_{t_i} expands, if there exists a community $C_{t_\theta} \in G$, $t_\theta > t_i$, that is bigger than, similar to, and shares nodes with C_i:

$$expand(C_{t_i}) = 1 \ \ if \ \ \exists\, C_{t_\theta} \in G/$$
$$sim(C_{t_i},\ C_{t_\theta}) = 1, \frac{|V_{t_i} \bigcap V_{t_\theta}|}{|V_{t_i} \bigcup V_{t_\theta}|} > \lambda,\ |V_{t_i}| < |V_{t_\theta}|$$

— *split*: A community C_{t_i} splits, if there exists a set of communities $\zeta_{t_\theta} = \left\{C_{t_\theta}^{q_1}, ..., C_{t_\theta}^{q_\theta}\right\}$, $t_\theta > t_i$, where each community of ζ_{t_θ} is similar to and shares nodes with C_{t_i}:

$$split(C_{t_i}) = 1 \ \ if \ \ \exists\, \zeta_{t_\theta} = \left\{C_{t_\theta}^{q_1}, ..., C_{t_\theta}^{q_\theta}\right\} /$$
$$\forall C_{t_\theta}^{k_\theta} \in \zeta_{t_\theta},\ sim(C_{t_i}, C_{t_\theta}^{k_\theta}) = 1, \frac{|V_{t_i} \bigcap V_{t_\theta}^{q_\theta}|}{|V_{t_i} \bigcup V_{t_\theta}^{q_\theta}|} > \lambda$$

— *merge*: A community C_{t_i} is from a merge, if there exists a set of communities

$\zeta_{t_\gamma} = \left\{C_{t_\gamma}^{q_1}, ..., C_{t_\gamma}^{q_\gamma}\right\}$, $t_\gamma < t_i$, where each community of ζ_{t_γ} is similar to and shares nodes with C_{t_i}:

$$merge(C_{t_i}) = 1 \ if \ \exists \, \zeta_{t_\gamma} = \left\{C_{t_\gamma}^{q_1}, ..., C_{t_\gamma}^{q_\gamma}\right\} / \\ \forall C_{t_\gamma}^{k_\gamma} \in \zeta_{t_\gamma}, \ sim(C_{t_i}, C_{t_\gamma}^{k_\theta}) = 1, \ \frac{|V_{t_i} \bigcap V_{t_\gamma}^{q_\gamma}|}{|V_{t_i} \bigcup V_{t_\gamma}^{q_\gamma}|} > \lambda$$

— *stable*: A community C_{t_i} is stable at a given time, if there exists a community $C_{t_\theta} \in G$, $t_\theta > t_i$, identical to C_{t_i}:

$$stable(C_{t_i}) = 1 \ if \ \exists \, C_{t_\theta} \in G \ / \ V_{t_\theta} = V_{t_i}$$

2.7 Dhouioui et al. [13]

This paper focused on the domain of healthcare. Dhouioui et al. used an iterative similarity-based approach with independent community detection and matching. The similarity measure used is a content-based Jaccard. The authors also defined 5 events community may undergo:

— Continuing: The set of edges is the same at two.

— Splitting: The splitting of a single dynamic community present at time $t-1$ is divided into two different communities at time t. Therefore, an additional dynamic community appears.

— Merging: This event is observed when two community at time $t-1$ match to a single community at time t.

— Dissolving: A dynamic community is removed when it has not been observed for at least d consecutive time steps.

— Forming: Refers to the case where a community C_t is observed at time t and which does observed at time $t-1$. Consequently, a new dynamic community C_t is created.

It also defined the events an evolving individuals may undergo:

— Join: Join a community which is distinct to its home community due to specific

event.

— Disappear: A node is said to disappear when it is observed in any earlier community but is not found in next time steps.

— leave: Leave the home community to join another one or to leave definitively the network after a specific event.

— Appear: A node is considered as new appearing when it occurs only in the actual time step.

2.8 Goldberg et al. [17]

Goldberg et al. developed an algorithmic framework for studying the evolution of communities in social networks. They develop a linear regression system to predict the lifespan of a community based on structural features extracted from the early stage of the community. They find that community's properties such as size, intensity and stability are the most important features to predict its lifespan.

The approach identified evolutive chains of communities. Given a time evolving graph, community detection on each snapshot is executed using a chosen static algorithm (including overlapping ones). Any intersection based measure can be used to match communities between snapshots. The authors propose a strategy to find the best chains of evolution for each community: they define chain strength as the strength of its weakest link. As a result, all the maximal valid chains are constructed for the identified communities. A valid chain is considered maximal if it is not a proper subchain of some other valid chain.

2.9 Xu et al. [38]

Xu et al. proposed a method for community tracking using an adaptive evolutionary clustering framework to reveal temporal evolution of communities. The idea is to exploit the knowledge about the previously found clustering to find a clustering for the current time step that is similar to the previous clustering, and is still a good

clustering also for the data in the current time step.

The authors use the normalized cut spectral clustering approach by [39], and they incorporate temporal smoothness by adapting the input data for the chosen clustering algorithm based on the community structure found in the previous snapshot. Their method is as follows. The adjacency matrices of the snapshots are considered as realizations of a non stationary random process which allows to define an expected adjacency matrix for the current snapshot. Based on this expected matrix a *smoothed adjacency matrix* can be approximated that also takes into account the previous time step. The smoothed adjacency matrix is a convex combination of the smoothed adjacency matrix of the previous time step and the actual adjacency matrix of the current time step. The chosen clustering algorithm is then applied to the estimated smoothed adjacency matrix, thus incorporating temporal smoothness to stabilize the variation of the found clusters over time. It aims to find stable clusters over time by penalizing deviations from incremental static clustering.

2.10 Concluding remarks

In studying these approaches, we observed that most of them use a two-stage process. As approaches in [34], [2], [24], [33], [5], [19], [10], they begin by considering the dynamic network under investigation as a series of static snapshot graphs at different time points. Then,

1. First stage: Using a community detection algorithm, they identify community structures at each of these snapshots independently.
2. Second stage: a pair-wise comparison based on a similarity measure is employed to track groups of users and detect changes they may undergo.

We note that the authors use different metrics when they apply community tracking and also they focus on different challenges. For example, they used different community detection algorithms to identify groups of nodes within the social network at each time-stamp. Moreover, their matching rely on different similarity measures.

2.10. Concluding remarks

Some authors, such as Brodka et al, in [10] tested their approach with both overlapping and non-overlapping communities, while others tested on non-overlapping communities only. Going through all those state-of-the-art approaches, we also notice that different authors like [19], [34] and [33], specified algorithms for tracking communities over time, and defined events that communities can undergo between either consecutive or non-consecutive snapshots as well. While authors in [2], [10], [16], in their work, built event-based frameworks to discover and track critical events over time, no community tracking process was specified. Moreover, besides their focuses, they used various community detection algorithms, similarity measures, threshold selection methods, even implemented different tracking approaches, all these highlights show a fact that, in the domain of tracking communities in dynamic social networks, it lacks a benchmark for other researchers to discriminate. At the mean time, it also added plenty of possibilities to social network studies to attract researchers to contribute. More details about our observations for those approaches will be given in Chapter 4 when we discuss our work "A comparative study of different approaches for tracking communities in evolving social networks" published in International Conference on Data Science and Advanced Analytics 2017.

Chapter 3

Proposed approach for tracking communities

We introduced several advanced techniques which have been successfully used in social network studies in the previous chapter. Most of the frameworks mainly focus on detecting and analyzing critical events that happen to groups over snapshots. Existing algorithms for tracking community also leave us some interesting problems to explore and tackle. It is clear by going through the previous chapters that the field of tracking communities which has a great value in a wide variety of application domains though still continuing to pose challenges to researchers in various way. For example, authors Asur et al. [2], Greene et al. [19], Gliwa et al. [16] and Takkaffoli et al. [34] make pair-wise comparisons, replying on a "Jaccard-based" similarity measure, which only takes into consideration of the proportion of nodes shared. For instance, the communities C_1 and C_2 can be considered similar if they have at least 30% of their nodes in common. And also, approaches by Asur et al. [2], Gliwa et al. [16] and Brodka et al. [10] perform their comparisons between community structures at two consecutive timestamps. Such approaches, however, may fail to track, because nodes or communities may be present at non-consecutive times. We also notice, in the approaches by Asur et al. [2], Greene et al. [19], Gliwa et al. [16], Brodka et al. [10] and Lee et al. [24], two communities are considered to be similar if the score returned by their similarity measure is above a user-specified threshold. The threshold val-

ues are manually set, which is difficult to justify, non stable and often inconsistent. Moreover, to obtain appropriate threshold values, one need numerous times of experiments, which would be impossible in some big-data analysis.

The purpose of this chapter is to alleviate these limitations by introducing a new approach for modeling and tracking communities over time. In our approach, we adopted a two-stage process. We first independently detect community structures at each snapshot. Then, we compare communities discovered from the whole time period, no matter the communities are from consecutive timestamps or not, which gives our approach the ability to track them even they "disappear" at some time steps. It is worth noting that the spotlight of our approach is the similarity measure that can capture all of the "Nodes Quality", "Time Proximity" and "Content Similarity" (These three claims are defined at *Summary* and will be elaborated in details in Section 3.2.1). Then, according to the similarity threshold which is automatically generated based on the distribution of similarity values between all the communities observed, we generate community sequences to represent the evolutions. This is also how communities are tracked over time in our approach.

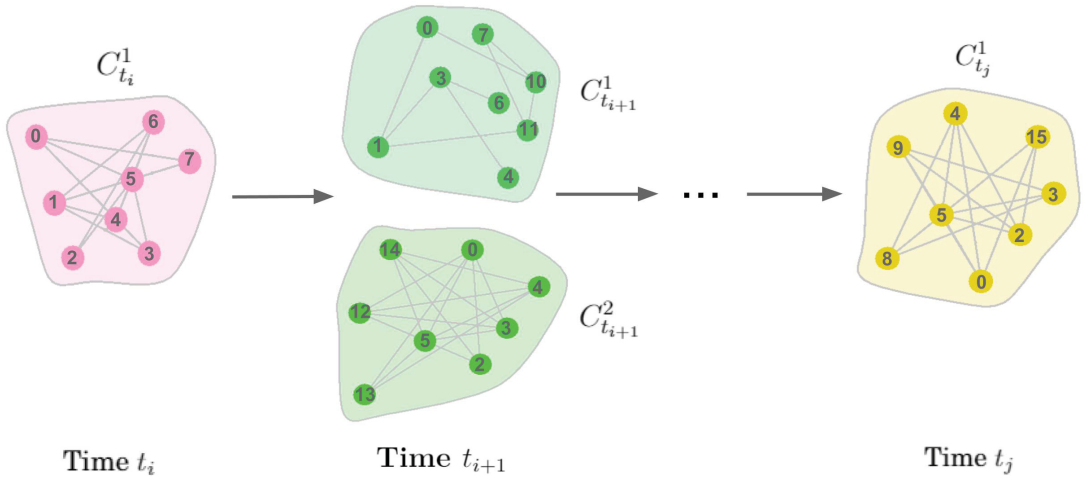
3.1 Problem Formalization

In the follows, we are going to do an abstract analysis of the evolution of community as an example to elaborate the shortcoming or misunderstanding caused by some existing approaches using the methods of diagrams. Therefore, no concrete calculation or formulation will be given. We will then introduce in the following section a new approach which could alleviate these limitations including the detailed processing steps. Also, to justify our proposed approach and reveal its properties against other algorithms, we implemented a comparative study with several popular approaches for tracking communities in dynamic social networks. Because of the greater length, this comparison analysis will be given in a following separated chapter.

We begin by introducing some notations we will use in this section.

— $C_{t_i}^n$ is the n_{th} community at time t_i;

Fig 3.1 – Evolution of Community $C^1_{t_i}$



— $N(C^n_{t_i})$ represents the list of nodes included in community $C^n_{t_i}$;
— We denote the *Jaccard Coefficient* or *Modified Jaccard Coefficient* between communities by $Jac(C^n_{t_i}, C^m_{t_j})$ (Eq 2.6);
— We use $d_{C^n_{t_i}}(N)$ to represent the sum of importance values of a set of nodes N in community $C^n_{t_i}$. Importance of node can be any measure which indicates member position within community. Here we use the degree, which indicates the number of edges of a node.

Let's look at Fig 3.1 which shows an example of the evolution of a community over time. First, community $C^1_{t_i}$ is detected at time t_i, it has 8 nodes whose numbers range from 0 to 7. Then two communities $C^1_{t_{i+1}}$ and $C^2_{t_{i+1}}$ are detected at time t_{i+1}. Respectively, $C^1_{t_{i+1}}$ and $C^2_{t_{i+1}}$ have 10 and 8 nodes, and 6 nodes and 5 nodes in common with $C^1_{t_i}$. For the better visualization, in our graph, community $C^1_{t_{i+1}}$ and $C^2_{t_{i+1}}$ will not be overlapped even they have common nodes. At time t_j, community $C^1_{t_j}$ is detected containing 8 nodes, 5 of which are in common with the original community $C^1_{t_i}$.

Tab 3.1 gives the data for the degree of each node in each community. Since from Fig 3.1, $C^1_{t_i}$ is the original community, which can be considered as start point of the evolution in our hypothesis. The problem formalization we are going to demonstrate

Tab 3.1 – Degree for Nodes in Each Community

(a) Degree for Nodes in $C^1_{t_i}$

N_0	N_1	N_2	N_3	N_4	N_5	N_6	N_7
3	4	2	3	4	7	3	2

(b) Degree for Nodes in $C^1_{t_{i+1}}$

N_0	N_1	N_3	N_4	N_6	N_7	N_{10}	N_{11}
2	3	3	1	1	2	3	3

(c) Degree for Nodes in $C^2_{t_{i+1}}$

N_0	N_2	N_3	N_4	N_5	N_{12}	N_{13}	N_{14}
5	3	4	4	7	4	3	4

(d) Degree for Nodes in $C^1_{t_j}$

N_0	N_2	N_3	N_4	N_5	N_8	N_9	N_{15}
3	4	3	4	7	3	3	3

below will revolve around the evolution of community $C^1_{t_i}$. For the convenience of visualization and to keep well track of the original members from the original community, in the four subtables from Tab 3.1, we colored all the nodes in $C^1_{t_i}$ (N_0 to N_7) in pink.

Next, we begin to track community $C^1_{t_i}$ conceptually by following different principles introduced in Chapter 2. First, if we try to simulate how the principles proposed by author Asur et al. [2], Greene et al. [19], Gliwa et al. [16] and Takkaffoli et al. [34] tracking communities, according to the similarity measure they used, they will consider community $C^1_{t_{i+1}}$ at time t_{i+1} as the continuation of the original community rather than $C^2_{t_{i+1}}$, just because $C^1_{t_{i+1}}$ has more common nodes with $C^1_{t_i}$, which refers to $Jac(C^1_{t_i}, C^1_{t_{i+1}}) > Jac(C^1_{t_i}, C^2_{t_{i+1}})$. But if we take a close look at the inner structure of these communities, we can notice common members of community $C^2_{t_{i+1}}$ and $C^1_{t_i}$ are more densely connected with other members inside community, it can be proven

by comparing the degrees of the pink nodes in Tab 3.1b and Tab 3.1c, we can see, $d_{C^1_{t_{i+1}}}(N(C^1_{t_i} \bigcap C^1_{t_{i+1}})) = 12$, but $d_{C^2_{t_{i+1}}}(N(C^1_{t_i} \bigcap C^2_{t_{i+1}})) = 23$. Please note that, in our studies we consider the quality of group members can be described by the member position within the community, in this case, we used node's degree. In other words, we can say, from our observation, from the original community $C^1_{t_i}$, most of the members have more important social positions are better preserved in $C^2_{t_{i+1}}$. In conclusion, from this example, we can briefly understand a content-based similarity measure can lead us to a wrong direction in terms of tracking communities.

In this paragraph we will demonstrate another issue which is neglected by most of the approaches, the time proximity between communities when we match them. First, we assume a tracking method can track communities over a non-consecutive timestamps. Based on that, $C^2_{t_{i+1}}$ and $C^1_{t_j}$ are equally eligible to be the continuation of the original $C^1_{t_i}$. Because both of them have the same common members with $C^1_{t_i}$ and their common members have approximately equal importances inside of each community ($d_{C^2_{t_{i+1}}}(N(C^1_{t_i} \bigcap C^2_{t_{i+1}})) \approx d_{C^1_{t_j}}(N(C^1_{t_i} \bigcap C^1_{t_j}))$). So a question is raised on, which would be the best candidate? For example, there is a social group allied by a current affair topic at January 2000, another similar group which has kept most of the core members is found at February 2000, compared to another group found at 2002 but doesn't have any more advantages other than the previous one, commonly, we would consider the one which is closer to the original community on the time line can better explain the course of topic. Unfortunately, a lot of approaches tracking communities over non-consecutive timestamps like [34], [19], [33], do not take the time proximity into consideration.

In conclusion, we comprehend the bias of some existing approaches through the examples given above, which motivates us to develop a new approach aiming at alleviating those limitations and accurately tracking and modeling communities.

3.2 Proposed Approach

3.2.1 Similarity Measure

Tracking community on off-line social networks is highly dependent on the similarity measure used. The most important component of the approach we proposed is a new similarity measure as well. As we talked about the motivation and goal of our work in the previous section. So in this section, we will mainly focus on developing the proposed similarity measure. First, let's define some notions that will be used below:

— Basically, a very important metric that can intuitively show us two communities are similar or not is their content similarity. Here, we denote it by the famous Jaccard Coefficient:

$$Jac(C_{t_i}^k,\ C_{t_{i+1}}^j)\ =\ \frac{|V_{t_i}^k\ \bigcap\ V_{t_{i+1}}^j|}{|V_{t_i}^k\ \bigcup\ V_{t_{i+1}}^j|} \tag{3.1}$$

— If community C_{t_i} and C_{t_j} have intersections, let's note nodes importances $NI(C_{t_i},\ C_{t_j})$, namely the contribution of their common members in C_{t_i}, as below:

$$NI(C_{t_i},\ C_{t_j}) = \frac{\sum_{x\ \in\ (C_{t_i}\ \bigcap\ C_{t_j})} d_{C_{t_i}}(x)}{\sum_{x\ \in\ (C_{t_i})} d_{C_{t_i}}(x)} \tag{3.2}$$

Here we use $d_{C_{t_i}}(x)$ to represent the contribution of node x in community C_{t_i}.

According to the definition of $NI(C_{t_i},\ C_{t_j})$, for two communities C_{t_i} and C_{t_j}, there are two formulations $NI(C_{t_i},\ C_{t_j})$ and $NI(C_{t_j},\ C_{t_i})$ can represent their common nodes contributions in two communities respectively. Since we are likely to give high NI score to two communities, which their common members have important social positions in both of them. In other words, we give preference to communities having similar important members when we match them. For this purpose, we calculate the average nodes quality using the

harmonic mean between $NI(C_{t_i},\ C_{t_j})$ and $NI(C_{t_j},\ C_{t_i})$:

$$H(C_{t_i},\ C_{t_j}) = \frac{2\ \cdot\ NI(C_{t_i},\ C_{t_j})\ \cdot\ NI(C_{t_j},\ C_{t_i})}{NI(C_{t_i},\ C_{t_j})\ +\ NI(C_{t_j},\ C_{t_i})} \tag{3.3}$$

— Clearly time stamps should play an important role in determining similarity, we wish the closer communities can be considered more relevant than further communities under the same condition, $T(C_{t_i},\ C_{t_j})$ represents the time proximity of community C_{t_i} and C_{t_j}, defined as below,

$$T(C_{t_i},\ C_{t_j}) = \frac{1}{e^{|\ t_i - t_j\ |}} \tag{3.4}$$

where we use an exponential function to incorporate the decaying effect of time lapse between the communities. The time granularity can be user-specified.

By combining all the metrics above, we propose a new similarity measure for capturing all of the "Content Similarity", "Nodes Quality" and "Time Proximity" has the range $(0, 1)$:

$$Sim(C_{t_i},\ C_{t_j}) = Jac(C_{t_i},\ C_{t_j})\ \cdot\ H(C_{t_i},\ C_{t_j})\ \cdot\ T(C_{t_i},\ C_{t_j}) \tag{3.5}$$

One might say this measure is slightly "strict" for non identical groups, because even if the two communities differs even by only one node, the NI part will decrease for not having all nodes and their social positions. But using the member position within the community calculated on the basis of users relations makes our measure to focus not only on the nodes, but also on the edges (relations), this will yield a great advantage over the methods which consider only member's overlapping for tracking.

3.2.2 Similarity Threshold Selection

In our methodology, we match community from different time stamps to accomplish tracking. When we compare them, there is a similarity threshold that we need to determine. In certain cases this threshold is set manually [2], [19], [9], [16]; in others it is determined by using semantic analysis of the network [34].

To determine the threshold automatically that yields the best similarity between two detected communities, we followed a principled approach from [33]. First, we compare all the similarity values excluding the zero ones, let S be the set of all similarity values greater than zero, ordered from smallest to largest. We assume that values in S are continuous and monotone in the interval $(0, 1)$. We consider threshold λ separates high values from low values in S. In other words, the similarity threshold is the value that separates set S into two subsets. In order to separate set S, we assume that it follows a mixture of Gammas or Betas which appears to be appropriate due to their shape flexibility. For reasons of simplicity, we adopted an approach based on two Gaussians distribution. For this propose, we apply a k-means ($k = 2$) on S to initially separate set S into subsets S_1, S_2, where values in S_1 is lower than values in S_2. We assume that values in S_1 follow the normal distributions $N(\sigma_1, \mu_1)$, values in S_2 follow the normal distributions $N(\sigma_2, \mu_2)$, with σ_1 and σ_2 the standard deviations, μ_1 and μ_2 the mean scores in S_1 and S_2 respectively. Let f_1 and f_2 be the respective probability density functions of subsets S_1 and S_2. The threshold value that best separates the set S is the value λ that satisfies $f_1(\lambda) = f_2(\lambda)$. Using λ, we define the binary similarity of two communities as follows:

$$Sim(C_{t_i}, C_{t_j}) = \begin{cases} 1 & \text{if } Sim(C_{t_i},\ C_{t_j}) > \lambda \\ 0 & \text{otherwise} \end{cases} \tag{3.6}$$

3.2.3 Tracking Communities Over Time

The objective of this study is to track communities over time in dynamic social networks, where several issues need to be addressed. For example, in order to detect community evolutions, a community matching strategy is applied. The set of communities extracted by a community mining algorithm at a given snapshot needs to be matched to the communities at previous snapshots based on their similarity. In general, the strategy of finding the optimal match between communities at different time steps will assume a zero-to-one or one-to-one mapping between nodes in the two communities – which will not readily support the identification of dynamic events such as community merging and splitting. A community may be similar to several communities at the same time. Thus, a simple greedy matching algorithm such as the one used in [36] cannot handle the case where the similarity of more than one communities to a previous community is the same. Also some approaches like [19] can be "strict" on communities which do not have members closely connected enough during the whole time window. They assume a community is "dissolved" if it has disappeared for more than 2 time stamps in the observation period.

We propose a matching algorithm that maximizes the pair-wise similarity over all selected matches, which allows for many-to-many mappings between communities across different time steps. Since our similarity measure introduced above handles the issue of community's time proximity, it gives us the freedom to consider if this community is "dissolved" comprehensively using the pair-wise similarity, not just abandoning the information after two time steps. So under the constraint of the similarity measure, we will implement matching over the entire observation period. It means our method will be able to maximize the potential evolutions we could find. The overview of the entire process is provided below, and Algorithm 1 represents the process of tracking a particular community.

1. Initially, each community at snapshot 0 is considered as a newly formed community and a new sequence s is created for each of them. We use these sequences to bootstrap the process.

2. In iteration t, extract communities from graph using community detection

method, then these communities will be matched to the last community from each sequence s. For which a pair has the similarity value above a certain threshold, the community will be saved to the corresponding sequence. If there are no matches, create another dynamic sequence containing the newborn community. Note that the last community in each sequence doesn't have to come from time $t-1$, it could be any community from $t-2$, $t-3$ or earlier as long as in the duration between t_{last} and t, there's no similar community found for this sequence, that's why this tracking strategy can track communities from non-consecutive time steps.

3. Update the newest community for each dynamic community sequences.
4. Repeat from #2 until all time step graphs have been processed.

In the next Chapter, we will validate the proposed approach with other five well known existing approaches on three real datasets, and evaluate every approach according both quantity and quality of tracking.

Algorithm 1 Tracking communities over time

Input : A community C_{t_i}, threshold similarity λ, all the timestamps T
Output : S includes a set of s defining the evolution of C_{t_i}
Initialize $s \leftarrow \{C_{t_i}\}$ {*initialize s with community* C_{t_i}}
Initialize an empty array S
$t_j = t_{i+1}$
while $t_j \leq t_k$, with t_k the last time in T **do**
  $g_{t_j} \leftarrow$ communities found at t_j
  for s in S **do**
    $C_{t_x} \in s$, with t_x the last time stamp in s
    for $C_{t_j} \in g_{t_j}$ **do**
      if $sim(C_{t_x}, C_{t_j}) = 1$ **then**
        $s_{copy} = copy(s)$
        $s_{new} \leftarrow s_{copy} \bigcup \left\{C_{t_j}\right\}$ {*save community* C_{t_j} *in* s_{copy}}
      else
        $s_{new} \leftarrow \left\{C_{t_j}\right\}$ {*initialize a* s_{new} *for* C_{t_j}}
      end if
      $S \leftarrow S \bigcup \{s_{new}\}$
    end for
  end for
  $t_j = t_{j+1}$
end while

Chapter 4

Comparative study

In the domain of social networks, tracking the evolution of groups of users within social networks has attracted growing interest from researchers due to the wide variety of application domains. Tracking the evolution of groups of users is our biggest concern. After going through the state-of-the-art approaches introduced in Chapter 2, we noticed that different authors tackle these challenges in different ways. They used various community detection algorithms, similarity measures, threshold selection methods, even implemented different tracking strategies. From this, we can conclude that the different approaches may perform differently on the dynamic social networks under investigation. Also in the domain of tracking communities in dynamic social network, it lacks of benchmark for other researchers to discriminate which method is better for them. So we decide to make a high level survey of some existing tracking approaches and then do a comparative analysis for them. In our analysis, we compared the algorithms in two main situations: (1) when groups of users do not overlap and (2) when the groups overlap. The study was done on three different testbeds extracted from the DBLP, Autonomous System (AS) and Yelp datasets.

Tab 4.1 – Overview

Approach	Community detection	Type of communities	Similarity measure	Threshold setting	Tracking
Greene et al. [19]	Blondel modularity [6]	Disjoint	Eq 3.1	Manually set	Consecutive & non-consecutive
Takaffoli et al. [34]	Local community mining [12]	Disjoint	Eq 2.2	Automatically set	Consecutive & non-consecutive
Brodka et al. [10]	CPM [1]	Overlapped	Eq 2.3	Manually set	Consecutive
Tajeuna et al. [33]	Infomap [29]	Disjoint	Eq 2.5	Automatically set	Consecutive & non-consecutive
Proposed approach	Infomap [29] & CPM[1]	Disjoint & Overlapped	Eq 3.5	Automatically set	Consecutive & non-consecutive

4.1 Algorithms

The algorithms used to perform comparative studies are:

— Greene et al. [19], specified in Sec 2.2
— Takaffoli et al. [34], specified in Sec 2.3
— Brodka et al. [10], specified in Sec 2.4
— Tajeuna et al. [33], specified in Sec 2.6
— Proposed approach, specified in Chapter 3

Despite all the information we have given in Chapter 2 and 3, Table 4.1 presents a summary of the mainstream approaches. The first column identifies the different approaches. The second column indicates the community detection algorithm used by each, and the third column, the type of communities identified by the particular community detection algorithm. The fourth column indicates the similarity measure used to compare the communities at different timestamps. In the fifth column, the strategy used to set the similarity threshold is identified. Finally, the last column presents the tracking results, which specifies whether the approach can track communities evolving in a consecutive or non-consecutive way.

In Table 4.1, it is worth noting that the notion of non-consecutive evolving communities taken in [19] differs slightly from the one in [34] and [33]. For instance, Greene et al. [19] assume that a community dissolves if no observation of it is found after $d = 2$ consecutive timestamps. It will thus be impossible to recognize it if it later reappears, rendering their approach unable to discover communities that evolve non-consecutively, with an interval of $d > 2$ consecutive timestamps during which there is no observation. In [34] and [33], a community is assumed to be dissolved if

there is no observation of it after the last time-stamp of observation.

Here are the reasons why we choose these four algorithms (except for proposed one) out of six introduced in Chapter 2 to implement comparative analysis. Though the community detection algorithms and similarity measures for tracking communities are different, there are several resemblances in some of the approaches. For instance, the tracking principles used in [16] and [10] are equivalents, differing only in the similarity measure used for finding critical events (these hypotheses were used to track the evolution of communities over time). Moreover, Gliwa et al. [16] already made the comparison of their approach with that of Brodka et al. [10]. Therefore, rather than repeating both of these approaches, we will instead run the approach in [10] with the one in [19], where the Jaccard coefficient is fully used. In the work done by Asur et al. [2], no community tracking process was specified. All that was demonstrated in their work is how to discover critical events that an evolving community may undergo at consecutive timestamps. In other words, they developed an algorithm for tracking events over time. However, since the focus of this paper is not to detail the various methodologies used to identify critical events, we have decided to avoid implementing the approach in [2] which is hard to compare with others returning sequences of communities. For the above reasons, we will only present experimental comparisons of the approaches in [19], [34], [10], [33] and our proposed approach.

4.2 Experiments

4.2.1 Data Description

In this section we validate five algorithms on three real-world datasets: the seventh version of the DBLP dataset [1], the Autonomous Systems (AS) dataset [2] and the YELP dataset [3]. It worth noting that the three datasets come with only nodes and edges.

1. *http* : *//arnetminer.org/citation*
2. *http* : *//snap.stanford.edu/data/as.html*
3. *http* : *//www.yelp.ca/academic_dataset*

1. The DBLP dataset contains co-publications of authors. For each published paper, it contains the paper's title, the authors, the year, the publication venue, the index identification of the paper and the identifications of references to the paper. We built undirected, unweighted graphs between co-authors and cited authors in the fields of data mining and artificial intelligence from 2011 to 2016, taking each year as a snapshot. Authors are represented by nodes, and co-authorships by edges.
2. The AS dataset contains the daily communication network of whotalkstowhom from the Border Gateway Protocol logs. We built undirected, unweighted graphs on communication networks on a daily basis from 3 October 1999 to 2 January 2000, where each identifier is considered as a node and a relation between two identifiers is taken as an edge.
3. In the YELP dataset, there are three main objects: "Business", "Review" and "User", giving information on businesses reviewed by users. We focused on the "User" object: for each user having friend(s), we created undirected, unweighted edge(s) between this user and his or her friend(s). The graph construction is done on a monthly basis from August 2009 to July 2014.

Several methods exist for detecting communities in social networks. In this chapter, we use two community-detection algorithms on all the three datasets separately. One is *Infomap* [29], which can detect non-overlapping communities. Another one is *CPM* [15], used to detect overlapping communities. By testing the five approaches on these two kinds of community sets separately, we can find out how this property will effect the performance of tracking approaches. Due to these two detection methods detect various numbers of communities on the three datasets, so for the purpose of keeping appropriate numbers of communities to track, we chose different numbers of snapshots to apply *Infomap* and *CPM*. Table 4.2 contains the descriptions of the datasets used for testing overlapping and disjoint communities, respectively. The second and third columns present the total number of nodes per social network. The fourth and fifth columns show the number of snapshots we chose for each social network.

Tab 4.2 – Dataset Descriptions Overall.

	Nodes		Snapshots	
#Network	Disjoint	Overlapping	Disjoint	Overlapping
DBLP	181474	181474	6	6
AS	6505	6741	20	60
YELP	34276	3904	20	118

4.2.2 Procedure

To implement a tracking procedure, we adopted a two-step approach. We first identify overlapping communities using the *CPM* algorithm and non-overlapping communities using the Infomap algorithm. The different algorithms are then compared by calculating the scores of the Jaccard coefficient between pairs of communities at distinct timestamps. For the metrics given in Eq 2.2, Eq 2.3 and Eq 2.5 we compare each pair of communities at distinct timestamps as well. Using a mixture of two Gaussian probability density functions, we automatically extract [33] the optimal threshold that characterizes the suitable similarity between two communities in the overall dataset under investigation. Note that this is done separately for each of the types of scores obtained. Thus, for each score obtained by a metric, we automatically identify the suitable threshold. After determining the different thresholds, we run the different algorithms in [34], [33], [19], [10] and proposed algorithm to identify the sequences of communities that reflect the evolution of communities over time. With the use of two general criteria (described below), we evaluate the purity of the evolving communities obtained in the five different cases.

In summary, the following steps are processed to obtain the results given in details in the next subsections:

— **Step 1**: Using the *CPM* and *Infomap* algorithms, identify the overlapping and non-overlapping communities at each timestamp.

— **Step 2**: Using the Jaccard coefficient as given in [19], calculate the score

for pairs of communities at distinct timestamps. Repeat the process for the metrics given in Eq 2.2, Eq 2.3 and Eq 2.5.

— **Step 3**: For each of the scores obtained in **Step 2**, automatically identify the suitable threshold using a mixture of Gaussians, as explained in [33].

— **Step 4**: With the thresholds obtained in **Step 3**, run the proposed algorithm and corresponding algorithms given in [10], [19], [33] and [34] to track the communities over time.

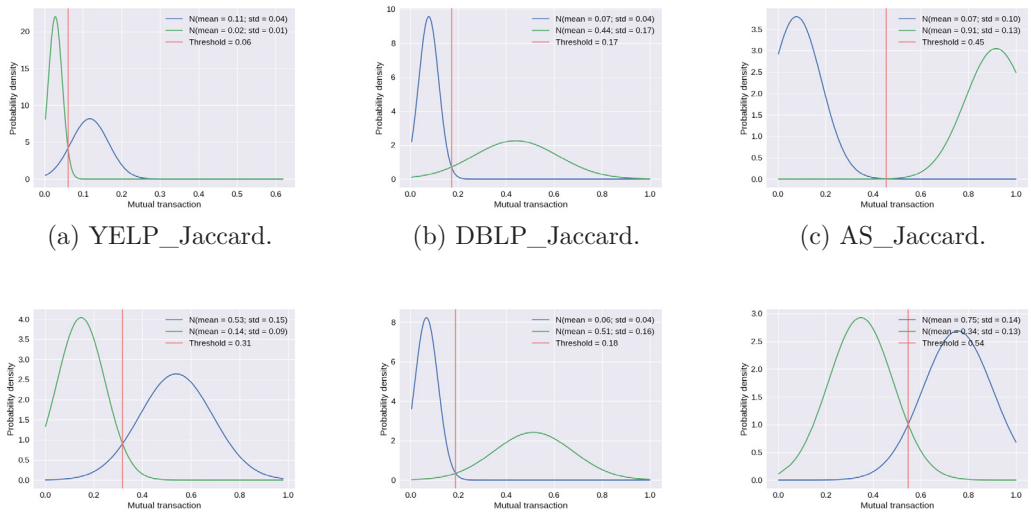
— **Step 5**: Calculate the purity (explained below) of the sequences obtained with all the different approaches.

4.2.3 Similarity Threshold Selection

When we compare communities to achieve tracking, there is a threshold that we need to respect, we adopted the same strategy we introduced at Section 3.2.2. As an example, consider the Jaccard similarity [20] (used in the methods of Greene et al. and Tajeuna et al.) and the mutual similarity (used in the method of Tajeuna et al.) implemented on overlapping communities from all the three datasets. Look at Fig 4.1, from the two Gaussians applied on the various distributions, we select as similarity threshold the mutual transition where the two Gaussians meet. The curves in Fig. 4.1b and Fig. 4.1e lie within $[0.15, 0.2]$, which indicates that the communities in DBLP change relatively quickly over time. On the other hand, the AS network is obviously more stable because the curve lies within $[0.4, 0.6]$, which indicates that the evolving communities observed here are more "alike". This is shown in Fig. 4.1c and Fig. 4.1f. For the similarity threshold selection of all the five tracking algorithms we are going to implement, we use the similarity threshold selecting using the mixture of two Gaussians.

4.2.4 Evaluation

For an objective comparison of the evolving communities (sequence of communities) obtained by different approaches, we look at the tracking results with respect



(a) YELP_Jaccard. (b) DBLP_Jaccard. (c) AS_Jaccard.

(d) YELP_Mutual. (e) DBLP_Mutual. (f) AS_Mutual.

Fig 4.1 – Threshold of similarity as the junction point of two Gaussian density curves.

to the original community of each evolution sequence. This means we only compare the sequences starting with the same community. We adopt a criterion to judge a approach: the lifespan of a evolution. For example, a community is found at January 2000, approach A can track its evolution until June 2000, then lose it after. However, approach B can successfully track the same community until December 2000, which means the last community can be considered as an evolution of the original community is found at December 2000. Obviously, as long as the process of matching is based on the resemblance criterion, the dynamic community sequence should be as long as possible, so that we can extract more properties to analyze community's behavior. According to this theory, apparently approach B compared to approach A has more advantages. So we add a coefficient which could represent the lifespan of sequences evolved from same original community found by all the tracking approaches. For this reason, we proposed α. Given an evolving community $S_C = C_{t_a} \rightarrow C_{t_{a+\eta}} \rightarrow ... \rightarrow C_{t_b}$, α is:

$$\alpha = \frac{t_b - t_a}{max(lifespans)} \tag{4.1}$$

Where "lifespans" is the lifespans of all the sequences start with C_{t_a} found by the four approaches.

In this thesis, based on our arguments above, we adopted two criteria to evaluate different tracking approaches, they are Average Pearson Correlation Coefficient (APCC) and Average Proportion of Nodes Persisting (APNP) respectively.

Average Pearson Correlation Coefficient (APCC)

The procedure is as follows: First, we adopt a general criterion based on the resemblance between each pair of selected communities in the evolving communities set. The global resemblance between two communities C_i and C_j is evaluated by the popular Pearson correlation coefficient [4], in which high value implies C_i, C_j are either overlapped or have similar changing pattern over time. It is defined as follows [4]:

$$\rho_{C_i,C_j} = \frac{(v_i - \bar{v_i}) \cdot (v_j - \bar{v_j})}{\|(v_i - \bar{v_i})\| \cdot \|(v_j - \bar{v_j})\|}, \tag{4.2}$$

where v_i and v_j are the corresponding transition probability vectors of communities C_i and C_j, respectively. They are the normalized number of shared nodes between C_i (C_j) and every remaining communities discovered over the whole time period, if there are N communities in total, the shape of the vector will be $[1, N]$. $\bar{v_i}$ and $\bar{v_j}$ are their respective mean values.

From equations 4.2 and 4.1, we calculate the average global resemblance for each pair of communities from an evolving community $S_C = C_{t_a} \rightarrow C_{t_{a+i}} \rightarrow ... \rightarrow C_{t_b}$ as the Average Pearson Correlation Coefficient (APCC), given as follows:

$$Avg\,(S_C) = \alpha \cdot \sum_{C_i \in S_C} \cdot \sum_{C_j \in S_C} \rho_{C_i,C_j} \tag{4.3}$$

4. Note that all values of the Pearson correlation coefficient are normalized within $[0, 1]$.

Average Proportion of Nodes Persisting (APNP)

The Average Proportion of Nodes Persisting (APNP) in an evolving community S_{V_C} ($S_{V_C} = \left\{V_{C_{t_a}},\ V_{C_{t_{a+i}}}, ...,\ V_{C_{t_b}}\right\}$ is the set of nodes corresponding to the sequence of community S_C), is an local inspection which reflects how the original members persist over time. It simply reveals the status extracted from different timestamps are correlated or not. We put coefficient α from equation 4.1 as a "bonus" into the equation for APNP, expressed as follows:

$$N_p(S_{V_C}) = \alpha \cdot \frac{\sum_{V_C \in S_{V_C}} \left| V_{C_{t_a}} \cap V_C \right|}{\left| V_{C_{t_a}} \right|} \tag{4.4}$$

Summary

An evolved community could be completely different compared to the original status, This may involve both the members of the community and its structures. Therefore, the criteria APNP is not a good enough candidate to evaluate the accuracy of evolution in this case, but criteria APCC, an inspection focusing on the global relevance of the evolution, gives us a chance to track community correctly even though a community is evolved to a completely different one. Evolutions of communities over time comprise what we refer to as community sequences. Note that different approaches could track community sequences in a different way. We compare how different methods track community sequences by considering both of the APCC and APNP values for each tracking method on each community sequence.

4.2.5 Experiment Based on Overlapping Communities Extracted by CPM

As a method for group extraction, *CPM* was utilized. The clique percolation method builds up the communities from k-cliques, which correspond to complete (fully connected) sub-graphs of k nodes. Here we choose $k = 4$. Table 4.3 shows

Tab 4.3 – Data Descriptions for Overlapping Communities used CPM.

Network	#avg_com	#com_size
DBLP	1983	8
AS	30	40
YELP	5	20

Tab 4.4 – Similarity Threshold for Each Method on Overlapping Communities in 4.3. Greene et al. using Jaccard coefficient, Takaffoli et al. using Modec similarity, Brodka et al. using the Inclusion similarity and Tajeuna et al. Mutual transition.

	Greene et al.	Takaffoli et al.	Brodka et al.	Tajeuna et al.	Proposed
DBLP	0.17	0.24	0.15 / 0.11	0.19	0.04
AS	0.45	0.39	0.40 / 0.41	0.55	0.08
YELP	0.06	0.08	0.10 / 0.03	0.32	0.01

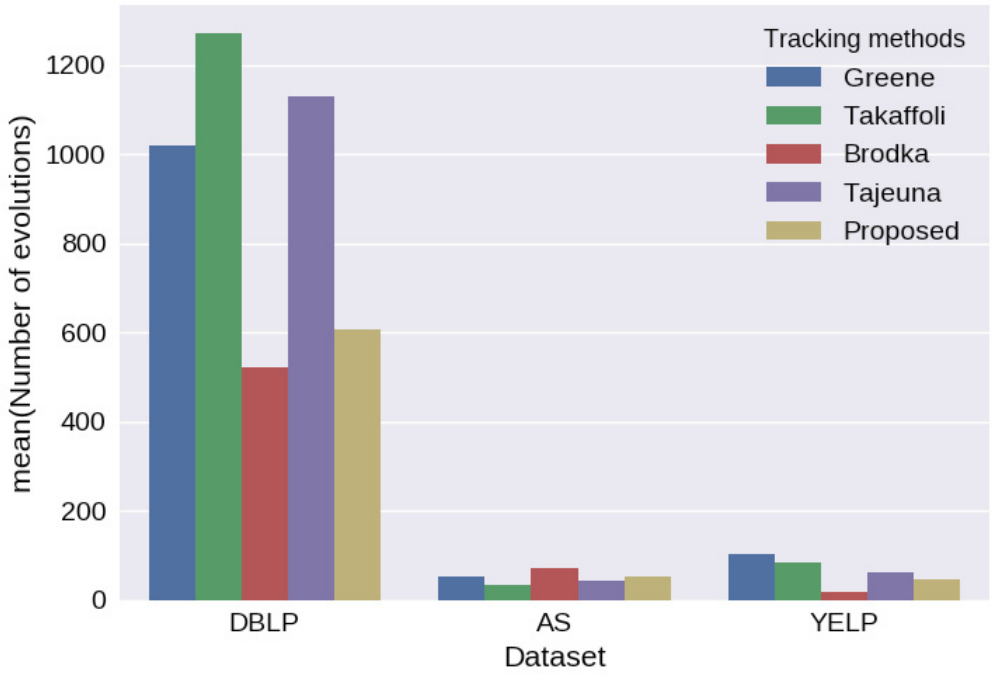
the description of the processed data: The first column shows the average number of communities detected by the *CPM* algorithm per snapshot, while the second column shows the average size of the groups per snapshot.

Table 4.4 gives the similarity threshold values for each of the five tracking methods. *Jaccard*, *Modec*, *Inclusion*, *Mutual* and the proposed similarity measure are the different similarity measures considered for these tracking methods. Note that for the similarity measure *Inclusion* given by Brodka et al. [10], for each pair of communities, we calculated both $I(C_i, C_j)$ and $I(C_j, C_i)$; this explains why two similarity threshold values are listed in the fourth column.

Quantity of Tracking

We observed that a number of communities undergo evolutions over time after implementing all five tracking approaches over these data streams which contain overlapping communities; we refer to these as evolving communities. To find sufficient number of evolving communities is very important to researchers to process

Fig 4.2 – Mean number of evolutions found on Overlapping Communities.



their ideas. Fig 4.2 shows the numbers of evolving communities found by different approaches on each dataset. We can then draw a number of conclusions on how these approaches compare with respect to the number of evolutions they find.

Let's look at Fig 4.2, X-axis represents different datasets, i.e. DBLP, AS, YELP separately. Y-axis indicates the number of evolving communities each approach finds. Different colors of the bar represent different approaches. Because different datasets have different sizes, this results in very different number of communities that will be found. In our case, DBLP dataset is much larger than AS and YELP, so, the first five columns corresponded to DBLP having the higher values than other columns. From the first five columns we observe that, for datasets with large graphs and small average size of overlapping communities like DBLP, the approaches of Greene et al., Takaffoli et al. and Tajeuna et al. can track a certain number of evolving commu-

nities, while that of Brodka et al. and the proposed approach can only find half of them. As a middle point, YELP includes overlapping communities of medium average size and medium-size graphs, which is the most common situation in real life. For this dataset, only the approach of Brodka et al. is not able to find sufficient number of evolving communities, and others are capable of. Therefore, we can say that Brodka's approach has less chance of receiving a good result in tracking small-size to medium-size communities when the quantity of tracking is of interest. The proposed approach has this letdown either, but it still can track more than Brodka's approach. We also noticed that, for datasets which include a large average size of overlapping communities such as AS, all the approaches perform well, especially approach Brodka et al..

Going through all the results, we can conclude that generally, the approaches of Greene et al., the approach of Tajeuna et al. and Takaffoli et al. are efficient enough to capture most potential evolutions over time. The Approach of Brodka et al. only performs well on big community-size dataset. However, the proposed approach sit at a middle point among them, no matter the properties of dataset, it can always track a certain amount of evolving communities, and it performs especially good on tracking big-size communities.

Quality of Tracking

From the previous paragraph, we know that different approaches are capable of tracking different communities or, we could say, different numbers of communities. Still, as mentioned before, the variety of tracking algorithms means that they could track the same community differently. In Fig 4.3, we show the quality of tracking of the five tracking algorithms. In each sub-figure, the X-axis presents those communities that all the approaches could successfully track and the Y-axis indicates the quality of the tracking algorithms on those communities. If we consider columns, each column has five values corresponding to how one community has been tracked by the five approaches. Each row presents the quality of a given tracking method over all the communities. Taking this into account, it is simple to estimate which

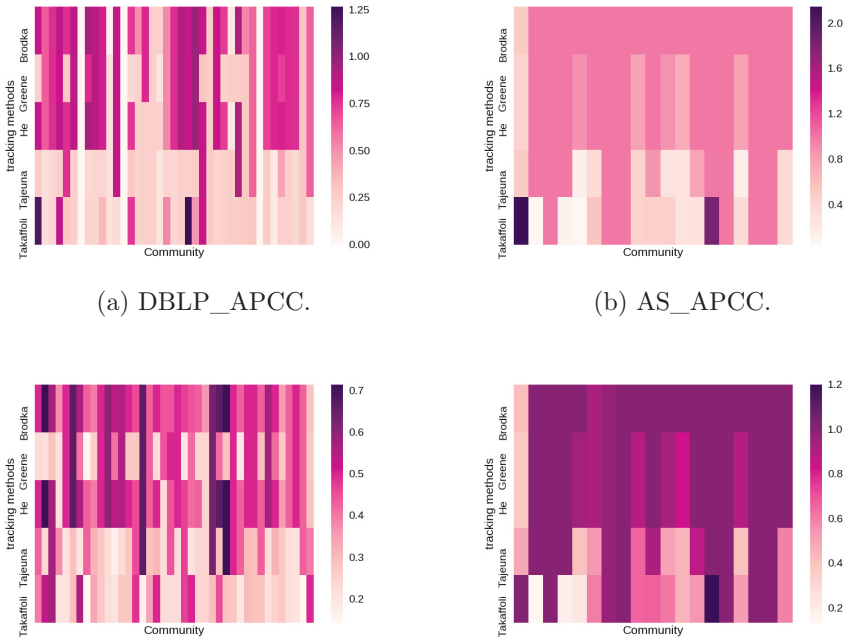
approach has better results by looking for the addresses of darker-colored cells. For this purpose, we look at the global performance using the statistical criteria defined in formulas 4.3 and 4.4 to define the quality of tracking, and make the comparison by looking at two specific criteria.

The numbers of communities selected as a starting point for an evolution which all the approaches can successfully track, used to plot the heatmap, are as follows:

— For DBLP, there are 431 communities.
— For AS, there are 19 communities.
— For YELP, there is only 1 community.

For the YELP dataset, due to the limited number of overlapping communities, there exists only one community that all five approaches can successfully track. Therefore, for this section only the demonstration on the DBLP and AS datasets is considered.

Fig 4.3 gives the heatmap for APCC and APNP. In Fig 4.3a and Fig 4.3b, each column illustrates the five APCC values for the evolutions of the same community tracked by the Brodka et al., Greene et al., Tajeuna et al. and Takaffoli et al. approaches. Fig 4.3c and Fig 4.3d presents the same illustration for APNP. As seen in Fig. 4.3a and Fig. 4.3c, the dataset DBLR contains big graphs and a large number of communities, so all the tracking approaches can successfully track a lot of communities. Looking at them, we observe that the darkest cubes appear in the row corresponding to the approach of Brodka et al., and proposed approach. Hence, we can conclude temporarily that, for datasets with small-size communities like DBLP, the approaches of Brodka et al. and proposed approach can track communities very well. For the dataset AS has big-size communities, shown in Fig. 4.3b and Fig. 4.3d, we can draw a similar conclusion. It is worth mentioning that approach of Greene et al. is also good at tracking big-size communities, Here it should be noted that the approach of Takaffoli et al. can track certain communities very well where other approaches fail, but the broad color range indicates that the performance of this approach is not stable enough. Overall, the approach of Brodka et al. and proposed approach are able to track a community very well in terms of tracking quality on



(a) DBLP_APCC.

(b) AS_APCC.

(c) DBLP_APNP.

(d) AS_APNP.

Fig 4.3 – In this heatmap, the darker the color of the cell, the higher the value. For (a) and (c) we removed the communities which all five approaches can track well (APCC and APNP are all above 0.3).

datasets which have overlapping communities.

4.2.6 Experiment Based on Non-overlapping Communities Extracted by Infomap

For the second experiment, the group detection algorithm Infomap was used on each snapshot. Table 4.5 contains the description of the processed data: The first column gives the average number of communities extracted by Infomap per snapshot, and the second shows the average size of the group. While Table 4.6 gives the similarity threshold values from each of the tracking methods on disjoint communities.

Tab 4.5 – Data Descriptions for Disjoint Communities.

Network	#avg_com	#com_size
DBLP	595	11
AS	129	17
YELP	222	16

Tab 4.6 – Similarity Threshold for Each Method on Disjoint Communities in 4.5. Greene et al. using Jaccard coefficient, Takaffoli et al. using Modec similarity, Brodka et al. using the Inclusion similarity and Tajeuna et al. Mutual transition.

	Greene et al.	Takaffoli et al.	Brodka et al.	Tajeuna et al.	Proposed
DBLP	0.11	0.15	0.10 / 0.11	0.15	0.02
AS	0.42	0.44	0.43 / 0.41	0.30	0.17
YELP	0.04	0.06	0.03 / 0.03	0.08	0.002

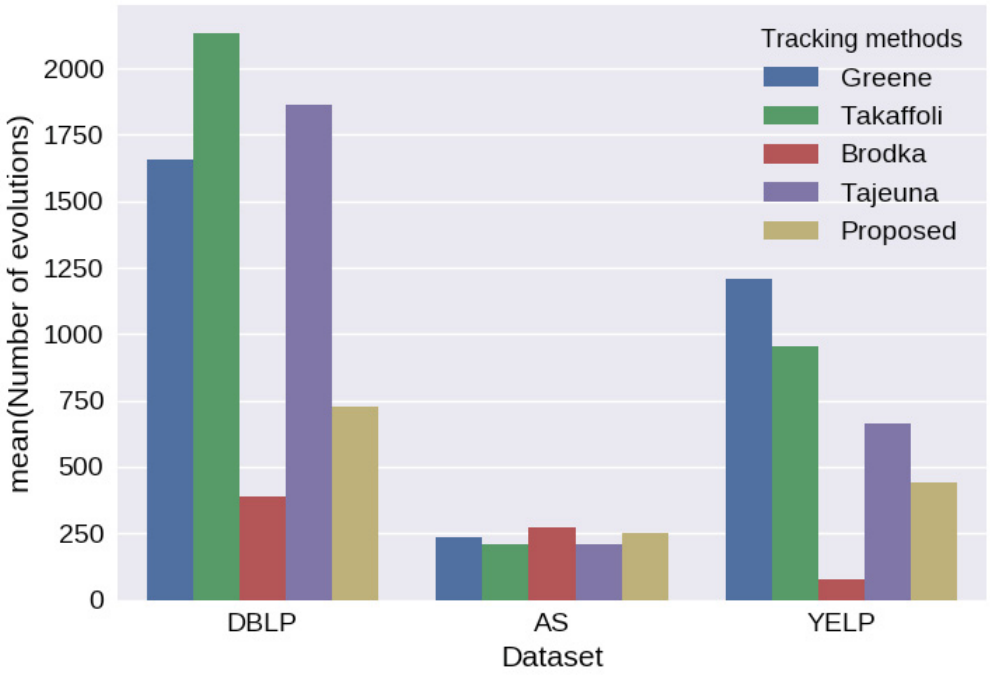
Quantity of Tracking

From the numbers of evolving communities found by different approaches on each dataset, shown in Fig 4.4, we obtain observations similar to those in Section 4.2.5. Brodka's approach has more chance of receiving a good audience in tracking only big-size communities when the quantity of tracking is of interest, and the approaches of Takaffoli et al., approach of Tajeuna et al. and Greene et al. are efficient enough to capture most potential evolutions over time. And for proposed approach, we can still achieve the similar observation, no matter the properties of dataset, it always can track a certain amount of evolving communities, and it performs especially good on tracking big-size communities.

Quality of Tracking

After implementing the five tracking methods on the three non-overlapping datasets, the community sequences obtained were processed by the same method used on overlapping communities. Fig. 4.5 is the heatmap of APCC and APNP. The interpretation of the heatmap has already been given in Sec. 4.2.5.

Fig 4.4 – Mean number of evolutions found on Overlapping Communities.



The numbers of communities selected as starting points for an evolution which all the approaches can successfully track, used to plot the heatmap, are as follows:

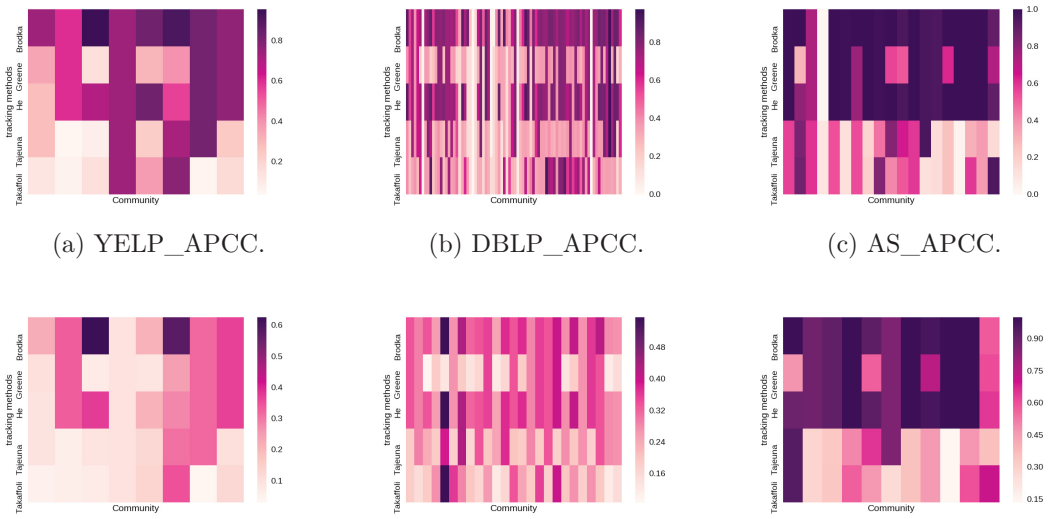
— For DBLP, there are 332 communities.

— For AS, there are 168 communities.

— For YELP, there are 8 communities.

Fig 4.5 contains the heatmap for APCC and APNP. In Fig 4.5a, Fig 4.5b and Fig 4.5c, each column gives the APCC for the evolutions of the same community tracked by the approaches of Brodka et al., Greene et al., Tajeuna et al., Takaffoli et al. and proposed approach. Fig 4.5d Fig 4.5e and Fig 4.5f give the same illustration for the APNP.

Again, similar conclusions can be reached. In all the sub-figures of Fig 4.5, most



(a) YELP_APCC. (b) DBLP_APCC. (c) AS_APCC.

(d) YELP_APNP. (e) DBLP_APNP. (f) AS_APNP.

Fig 4.5 – In this heatmap, the darker the color of the cell, the higher the value. For (b) we removed the communities which all four communities track well (APCC is above 0.4), for (e) we removed the community evolutions with APNP higher than 0.2. And for (c) and (f) we again removed the communities which all four approaches track well (APCC and APNP are all above 0.6).

of the darker cells are located in the row corresponding to the approach of Brodka et al. and proposed approach, according to the broad color range. We conclude that generally, the approach of Brodka et al. and proposed approach can track a community very well on datasets which have disjoint communities.

4.3 Result Analysis

In this section, we have presented a comparative study of approaches for tracking communities in time-evolving social networks in which communities may be either overlapping or disjoint. We introduced six existing popular tracking algorithms and tested five of them including proposed approach in our Experiment section.

For purposes of evaluation, we introduced two measures, the Average Pearson Correlation and the Proportion of Nodes Persisting, to evaluate the quality of the communities tracked. We performed the comparison study on real data from the DBLP, AS and YELP datasets (note that in our case, we consider the communities detected in dataset AS to be large-size communities, those from YELP to be medium-size, and those from DBLP to be small-size; for details see Table 4.3 and Table 4.5).

Our comparison reveals that all the approaches are capable of tracking overlapping and disjoint communities over time into sequences in which the set of communities shows good global resemblance (above the threshold of similarity). Generally, looking at both overlapping and disjoint communities, we observed that all the approaches are capable of tracking community evolutions very well. Normally approaches of Greene et al., Takaffoli et al. and Tajeuna et al. are capable of tracking a certain number of evolutions of communities over time. Approach of Brodka et al. and proposed approach can only find sufficient number of evolutions when tracking big-size communities, but achieve especially satisfying results with respect to APCC and APNP values on most dataset (in our experiment we only demonstrated 3 representative real datasets, so here we cannot popularize it on all datasets).

We conclude that when the dataset which will be dealt with has big-size communities, or, quality of tracking is focused, approach of Brodka et al. and proposed approach are the best choice, if the quantity of tracking is of interest, approaches of Greene et al., Takaffoli et al. and Tajeuna et al. are worth trying. In the future, we will focus on tracking and predicting critical events a community may undergo.

Conclusion

In our study, we proposed a novel statistical algorithm for modeling and tracking communities over time to deal with the main issues we have addressed in dynamic social networks. Briefly, c In our approach, we adopted a two-stage process. We first independently detect community structures at each snapshot. Then, in order to detect community evolutions, a communities matching strategy is applied. Communities extracted at a given snapshot will be matched to the communities at previous snapshots based on their similarity. The spotlight of our approach is the similarity measure which captures "Nodes Quality", "Time Proximity" and "Content Similarity". And a matching strategy allows for many-to-many mappings between communities across different time stamps.

Another important contribution of this thesis is a comparative study of different approaches for tracking communities over time in dynamic social networks. Since in the domain of tracking communities, authors of popular algorithms use various community detection algorithms, similarity measures, threshold selection methods, even implemented different tracking approaches, all these highlights show a fact that, in the domain of tracking communities in dynamic social networks, it lacks a benchmark for other researchers to discriminate. Also, it added plenty of possibilities to social network studies to attract researchers to contribute. So we decide to make a high level survey of some existing tracking approaches and then do a comparative analysis for them. In our analysis, we compared the algorithms in two kinds of community sets: (1) when groups of users do not overlap and (2) when the groups overlap. The study was done on three different testbeds extracted from the DBLP, Autonomous System (AS) and Yelp datasets. Our comparison reveals that all the approaches are capable

of tracking overlapping and disjoint communities over time. When the dataset has big-size communities, or, quality of community tracking is focused, the proposed approach and the approach of Brodka et al. and are the best choice, if the quantity of tracking is of interest, the approaches of Greene et al., Takaffoli et al. and Tajeuna et al. are worth trying.

The result of our work shows great prospect for the future of community tracking in the domain of dynamic social networks. To go further, we could investigate the detection and analysis of the critical events a community may undergo in order to predict when a change will happen to a community.

Bibliography

[1] Balázs Adamcsek, Gergely Palla, Illés J Farkas, Imre Derényi et Tamás Vicsek.
"CFinder: locating cliques and overlapping modules in biological networks".
Bioinformatics, 22(8):1021–1023, 2006.

[2] Sitaram Asur, Srinivasan Parthasarathy et Duygu Ucar.
"An event-based framework for characterizing the evolutionary behavior of interaction graphs".
ACM Transactions on Knowledge Discovery from Data (TKDD), 3(4):16, 2009.

[3] Zhifeng Bao, Yong Zeng et YC Tay.
"sonLP: social network link prediction by principal component regression".
Dans *Advances in Social Networks Analysis and Mining (ASONAM), 2013 IEEE/ACM International Conference on*, pages 364–371. IEEE, 2013.

[4] Jacob Benesty, Jingdong Chen, Yiteng Huang et Israel Cohen.
"Pearson correlation coefficient".
Dans *Noise reduction in speech processing*, pages 1–4. Springer, 2009.

[5] Sajid Yousuf Bhat et Muhammad Abulaish.
"HOCTracker: Tracking the evolution of hierarchical and overlapping communities in dynamic social networks".
IEEE Transactions on Knowledge and Data engineering, 27(4):1019–1013, 2015.

[6] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte et Etienne Lefebvre.
"Fast unfolding of communities in large networks".
Journal of statistical mechanics: theory and experiment, 2008(10):P10008, 2008.

[7] Yuri Boykov et Vladimir Kolmogorov.
"An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision".
IEEE transactions on pattern analysis and machine intelligence, 26(9):1124–1137, 2004.

[8] Piotr Bródka, Przemysław Kazienko et Bartosz Kołoszczyk.
"Predicting group evolution in the social network".
Dans *International Conference on Social Informatics*, pages 54–67. Springer, 2012.

[9] Piotr Bródka, Stanislaw Saganowski et Przemyslaw Kazienko.
"Group evolution discovery in social networks".
Dans *Advances in Social Networks Analysis and Mining (ASONAM), 2011 International Conference on*, pages 247–253. IEEE, 2011.

[10] Piotr Bródka, Stanisław Saganowski et Przemysław Kazienko.
"GED: the method for group evolution discovery in social networks".
Social Network Analysis and Mining, 3(1):1–14, 2013.

[11] Antoni Calvó-Armengol et Yves Zenou.
"Social networks and crime decisions: The role of social structure in facilitating delinquent behavior".
International Economic Review, 45(3):939–958, 2004.

[12] Jiyang Chen, Osmar Zaïane et Randy Goebel.
"Local community identification in social networks".
Dans *Social Network Analysis and Mining, 2009. ASONAM'09. International Conference on Advances in*, pages 237–242. IEEE, 2009.

[13] Zeineb Dhouioui et Jalel Akaichi.
"Tracking dynamic community evolution in social networks".
Dans *Advances in Social Networks Analysis and Mining (ASONAM), 2014 IEEE/ACM International Conference on*, pages 764–770. IEEE, 2014.

[14] Georgios Diakidis, Despoina Karna, Dimitris Fasarakis-Hilliard, Dimitrios Vogiatzis et George Paliouras.
"Predicting the evolution of communities in social networks".

Dans *Proceedings of the 5th International Conference on Web Intelligence, Mining and Semantics*, Page 1. ACM, 2015.

[15] Santo Fortunato.
"Community detection in graphs".
Physics reports, 486(3):75–174, 2010.

[16] Bogdan Gliwa, Piotr Bródka, Anna Zygmunt, Stanislaw Saganowski, Przemyslaw Kazienko et Jaroslaw Kozlak.
"Different approaches to community evolution prediction in blogosphere".
Dans *Advances in Social Networks Analysis and Mining (ASONAM), 2013 IEEE/ACM International Conference on*, pages 1291–1298. IEEE, 2013.

[17] Mark K Goldberg, Malik Magdon-Ismail, Srinivas Nambirajan et James Thompson.
"Tracking and Predicting Evolution of Social Communities.".
Dans *SocialCom/PASSAT*, pages 780–783. Citeseer, 2011.

[18] Benjamin H Good, Yves-Alexandre de Montjoye et Aaron Clauset.
"Performance of modularity maximization in practical contexts".
Physical Review E, 81(4):046106, 2010.

[19] Derek Greene, Donal Doyle et Padraig Cunningham.
"Tracking the evolution of communities in dynamic social networks".
Dans *Advances in social networks analysis and mining (ASONAM), 2010 international conference on*, pages 176–183. IEEE, 2010.

[20] Paul Jaccard.
"The distribution of the flora in the alpine zone.".
New phytologist, 11(2):37–50, 1912.

[21] Stephen C Johnson.
"Hierarchical clustering schemes".
Psychometrika, 32(3):241–254, 1967.

[22] Bryan Klimt et Yiming Yang.
"The enron corpus: A new dataset for email classification research".
Dans *European Conference on Machine Learning*, pages 217–226. Springer, 2004.

[23] Andrea Lancichinetti, Santo Fortunato et Filippo Radicchi.
"Benchmark graphs for testing community detection algorithms".
Physical review E, 78(4):046110, 2008.

[24] Pei Lee, Laks VS Lakshmanan et Evangelos E Milios.
"Incremental cluster evolution tracking from highly dynamic network data".
Dans *Data Engineering (ICDE), 2014 IEEE 30th International Conference on*, pages 3–14. IEEE, 2014.

[25] Jure Leskovec, Kevin J Lang et Michael Mahoney.
"Empirical comparison of algorithms for network community detection".
Dans *Proceedings of the 19th international conference on World wide web*, pages 631–640. ACM, 2010.

[26] Douglas A Luke et Jenine K Harris.
"Network analysis in public health: history, methods, and applications".
Annu. Rev. Public Health, 28:69–93, 2007.

[27] David M. S. Rodrigues.
"Identifying news clusters using Q-analysis and modularity".
Dans *European Conference on Complex Systems 2013*.
2013.

[28] Ryan A Rossi, Brian Gallagher, Jennifer Neville et Keith Henderson.
"Modeling dynamic behavior in large evolving graphs".
Dans *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 667–676. ACM, 2013.

[29] Martin Rosvall et Carl T Bergstrom.
"Maps of random walks on complex networks reveal community structure".
Proceedings of the National Academy of Sciences, 105(4):1118–1123, 2008.

[30] Purnamrita Sarkar et Andrew W Moore.
"Dynamic social network analysis using latent space models".
ACM SIGKDD Explorations Newsletter, 7(2):31–40, 2005.

[31] David B Skillicorn, Quan Zheng et Carlo Morselli.
"Spectral embedding for dynamic social networks".
Dans *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 316–323. ACM, 2013.

[32] Etienne Gael Tajeuna.
"*Suivi des communautés dans les réseaux sociaux dynamiques*".
Thèse de doctorat, Université de Sherbrooke, 2015.
unpublished thesis.

[33] Etienne Gael Tajeuna, Mohamed Bouguessa et Shengrui Wang.
"Tracking the evolution of community structures in time-evolving social networks".
Dans *Data Science and Advanced Analytics (DSAA), 2015. 36678 2015. IEEE International Conference on*, pages 1–10. *International Conference on Data Science and Advanced Analytics (DSAA)*, 2015.

[34] Mansoureh Takaffoli, Justin Fagnan, Farzad Sangi et Osmar R Zaïane.
"Tracking changes in dynamic information networks".
Dans *Computational Aspects of Social Networks (CASoN), 2011 International Conference on*, pages 94–101. IEEE, 2011.

[35] Mansoureh Takaffoli, Reihaneh Rabbany et Osmar R Zaïane.
"Community evolution prediction in dynamic social networks".
Dans *Advances in Social Networks Analysis and Mining (ASONAM), 2014 IEEE/ACM International Conference on*, pages 9–16. IEEE, 2014.

[36] Mansoureh Takaffoli, Farzad Sangi, Justin Fagnan et Osmar R. Zaïane.
"MODEC - Modeling and Detecting Evolutions of Communities".
Dans *ICWSM*, 2011.

[37] Xuning Tang et Christopher C Yang.
"Detecting social media hidden communities using dynamic stochastic blockmodel with temporal Dirichlet process".
ACM Transactions on Intelligent Systems and Technology (TIST), 5(2):36, 2014.

[38] Kevin S Xu, Mark Kliger et Alfred O Hero.
"Tracking communities in dynamic social networks".
Dans *International Conference on Social Computing, Behavioral-Cultural Modeling, and Prediction*, pages 219–226. Springer, 2011.

[39] Stella X. Yu et Jianbo Shi.
"Multiclass Spectral Clustering".
Dans *ICCV*, 2003.

[40] Yu Zhang, Zhaohui Wu, Huajun Chen, Hao Sheng et Jun Ma.
"Mining Target Marketing Groups From Users' Web of Trust on Epinions.".
Dans *AAAI Spring Symposium: Social Information Processing*, Page 116, 2008.